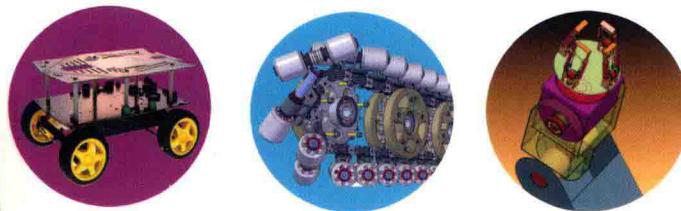


移动机器人导航与控制 算法设计

YIDONG JIQIREN DAOHANG YU KONGZHI
SUANFA SHEJI

张元良 著



华中科技大学出版社
<http://www.hustp.com>

内 容 提 要

本书主要论述了移动机器人导航系统和路径跟踪控制算法的设计方法。首先,基于常用的移动机器人导航系统的原理、所选用的民用单频 GPS 接收机数据的特点和航迹推算导航系统数据的特点,提出了三种 GPS 和航迹推算导航综合导航系统信息融合算法。这些算法利用 GPS 与航迹推算导航系统具有互补的特性,在解算负担小的基础上,能够很好地融合民用单频 GPS 接收机和航迹推算系统的导航信息,为室外移动机器人提供长期稳定、准确的导航信息。其次,介绍了移动机器人路径规划应用的蚁群算法。针对传统蚁群算法容易陷入早熟、收敛速度慢和生成路径转折多等问题,提出了相应的改进措施。改进后的蚁群算法在提高搜索速度的基础上,扩大了搜索范围,并且能够提供转折较少的路径。最后,基于模型算法控制方法和神经网络的模型算法控制方法为移动机器人设计了路径跟踪控制算法。

图书在版编目(CIP)数据

移动机器人导航与控制算法设计/张元良著. —武汉: 华中科技大学出版社, 2016. 12

ISBN 978-7-5680-2448-8

I. ①移… II. ①张… III. ①移动式机器人-研究 IV. ①TP242

中国版本图书馆 CIP 数据核字(2016)第 303591 号

移动机器人导航与控制算法设计

Yidong Jiqiren Daohang yu Kongzhi Suanfa Sheji

张元良 著

策划编辑: 王 剑

责任编辑: 张少奇

封面设计: 原色设计

责任校对: 张会军

责任监印: 朱 珍

出版发行: 华中科技大学出版社(中国·武汉) 电话: (027)81321913

武汉市东湖新技术开发区华工科技园 邮编: 430223

录 排: 武汉楚海文化传播有限公司

印 刷: 虎彩印艺股份有限公司

开 本: 710mm×1000mm 1/16

印 张: 8.5

字 数: 170 千字

版 次: 2016 年 12 月第 1 版第 1 次印刷

定 价: 38.00 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线: 400-6679-118 竭诚为您服务
版权所有 侵权必究

前　　言

近些年来,移动机器人的应用领域越来越广泛。稳定和准确的导航系统是移动机器人完成任务的重要前提。GPS是最常用的室外导航系统之一,随着SA(selective availability)政策的取消,民用GPS的精度得到了大幅度的提高。但是民用单频GPS的误差依然在10米左右,使得其无法单独为移动机器人提供导航服务。航迹推算导航系统具有短期精度高的特点。但是其误差会不断累积。GPS与航迹推算导航系统具有互补的特性,因此将二者进行融合组成综合导航系统是解决室外移动机器人导航问题的很好的方法。在得到了稳定的和准确的导航信息后,在很多的应用中常常需要进行路径规划。在栅格地图中进行机器人路径规划的方法有很多,各有其优缺点。如何能够在短时间内得到最优的路径是解决问题的关键,而路径跟踪控制问题是移动机器人控制领域的研究热点。轮式移动机器人具有非完整特性,即有轮子与地面为纯滚动而无滑动的约束条件。在设计轮式移动机器人路径跟踪控制系统时要考虑这些约束条件。

本书基于作者多年从事移动机器人导航和控制系统的研究成果,并参考了国内外相关的科技文献与资料,论述了基于民用单频GPS接收机数据特点所设计的GPS/航迹推算综合导航系统,基于改进的蚁群算法设计的移动机器人路径规划算法,以及基于模型算法控制方法与神经网络模型算法控制方法设计的轮式移动机器人路径跟踪控制系统。本书内容主要包括概述、GPS/DR室外移动机器人导航系统、移动机器人的路径规划、移动机器人路径跟踪控制共四章。在本书出版之际,向本书所引用的文献与资料的原作者以及支持和帮助本书出版的专家和同仁们表示由衷的感谢。

在本书的编著过程中,淮海工学院文西芹教授、赵明光教授对本书进行了审阅并提出了许多宝贵意见,淮海工学院刘青老师参加了资料收集和书稿整理,在此深表感谢。

特别感谢江苏省重点建设学科“机械制造及其自动化”项目、江苏省高校自然科学研究面上项目(13KJB510003)、青蓝工程和连云港市第五期“521高层次人才培养工程”项目对本书的经费支持。

由于作者水平有限,加之时间紧迫,书中的疏漏和不当之处在所难免,敬请读者批评指正,并提出宝贵意见。

张元良

2016年6月

目 录

第 1 章 概述	(1)
第 2 章 GPS/DR 室外移动机器人导航系统	(7)
2.1 常用的机器人导航系统	(7)
2.2 GPS 定位导航	(12)
2.3 导航信息融合算法的设计	(29)
第 3 章 移动机器人的路径规划	(52)
3.1 A * 算法	(52)
3.2 蚁群算法	(56)
第 4 章 移动机器人路径跟踪控制	(77)
4.1 数学基础	(77)
4.2 模型算法控制(MAC)	(82)
参考文献	(122)

第1章 概述

智能移动机器人是一个集环境感知、动态决策与规划、行为控制与执行等多功能于一体的综合智能系统^[1]。智能移动机器人系统中包含了电子技术、计算机技术、传感器技术、信息处理技术、自动化控制系统甚至人工智能系统等多学科的研究成果,是机电一体化的最高成就,也是当今研究的热点之一,在工业和农业生产中有着重要的地位。随着机器人性能不断地完善和功能不断地拓展,移动机器人的应用范围也大为扩展。现在,智能移动机器人在工业、农业、医疗、服务等行业中得到了广泛的应用,而且在城市安全、国防和空间探测领域中的有害与危险场合得到很好的应用。因此,移动机器人技术已经得到世界各国相关研究工作者的普遍关注。近些年来室外移动机器人得到了越来越广泛的关注和研究,其应用非常的广泛。移动机器人的研究始于20世纪60年代末期。斯坦福研究院(SRI)的Nils Nilssen和Charles Rosen等人,在1966年至1972年中研发出了取名Shakey的自主移动机器人^[2],其目的是研究应用人工智能技术在复杂环境下机器人系统的自主推理、规划和控制。室外移动机器人可以去执行很多危险的、枯燥的工作,在工业、农业、医疗、娱乐等方面有着很强的实用价值和应用前景。例如,“精确农业”是近年来国际上农业科学的研究热点领域,而农业移动机器人被认为是精确农业体系中不可缺少的一部分^[3]。在所有的应用中,能够准确地确定当前机器人的位姿以及提供准确的、长期可靠的导航信息成为了机器人完成其功能与任务的基本要求。

导航定位有多种技术途径。例如对于船舶导航技术来说,古老的海上定位方法是利用六分仪观测天体或陆地标志来推算船位,称为天文定位或地物定位。这些方法所测定的船位误差往往超过1海里。现代的导航定位系统有无线电定位系统、惯性导航系统、声学导航系统、卫星导航系统和综合导航系统等。无线电导航定位系统包括近程高精度定位系统和中远程导航定位系统。最早的无线电导航定位系统是20世纪初发明的无线电测向系统。20世纪40年代起,人们研制了一系列双曲线无线电导航系统,如美国的“罗兰”和“欧米加”,英国的“台卡”等。惯性导航的原理是利用运动物体的加速度和位置坐标之间的数学关系,确定物体相对于地面的位置。比较常用的惯性导航系统主要包括惯性平台(安装了高精度陀螺仪、加速度计等仪器)、电子计算机和辅助设备。声学导航是利用水下声波的产生、传播、接收等特性进行导航定位^[4]的。多普勒导航仪就属于声学导航范围。其换能器向海底发射超声波脉冲,并接收从海底和水中散射层反射的回波,在脉冲频率分

析器中把反射脉冲与发射脉冲的频率进行比较,根据反射脉冲的多普勒频移就可得知运动方向和距离。卫星导航系统是发展潜力最大的导航系统。卫星导航是把卫星作为信号发射台,配置在移动机器人上的接收机接收卫星发射的信号,使用卫星导航接收机自动测量信号的多普勒频移,计算出移动机器人的地理位置。1964年,美国推出了世界上第一个卫星导航系统——海军卫星导航系统,又称子午仪卫星导航系统^[5]。目前,该系统已成为使用最为广泛的船舶导航系统。随着小型计算机和微处理机的出现,组合导航系统得到了发展。采用组合导航系统可以发挥各类分系统的优点来弥补各自的不足之处,通过特殊数字模型处理,定位精度和可靠性都得到了显著的提高。可以说是移动机器人综合导航系统的发展方向。

以上介绍的导航系统都有各自的优点,同时也都具有各自的不足之处。对于利用航迹推算系统进行移动机器人导航来说,该系统能够提供短时精确的导航定位信息,但是由于导航信息经过积分而产生,定位误差会随时间的增长而增大,所以长期导航精度差。也就是说航迹推算导航系统无法长期的单独为移动机器人提供精确的导航信息,其误差随着时间的增加而增加,而且是没有上限的。GPS 全球卫星定位导航系统具有全球覆盖、不受地球上任何地域的限制、全天候不受天气等条件的影响等优势。现今 GPS 得到了广泛的应用,如定位、导航、绘图、测量和授时等。对于室外导航应用来说, GPS 可以提供三维的位置和速度信息。在适宜的条件下, GPS 可以提供长期连续和相对准确的导航信息。到目前为止关于移动机器人和自主式车辆基于 GPS 的导航系统得到了广泛的研究^[6,7]。Seo 等为室外移动机器人提出了一种基于 GPS 和惯性导航系统(INS)的室外综合导航算法^[8]。文献[9]利用低精度的惯性传感器和 GPS 为移动机器人提供导航信息。在该应用中通过适当的抑制 INS 的误差可以提高该低精度 INS 的准确性。在文献[10]中, Adusumilli 等提出了一种基于随机森林回归模型的 INS 和 GPS 导航信息融合算法来为移动机器人提供连续和准确的导航信息。在 GPS 无法正常工作的情况下,该随机森林回归模型可以进行 INS 误差估算,从而保持了导航信息的连续性。文献[11]的作者提出了一种低成本的农用机械导航系统。该导航系统包括三个陀螺仪和两个罗盘,利用最小二乘法来估算系统倾角,同时通过与 GPS 的信息融合来进行方向和位置的估算。文献[12]将视觉导航系统和 GPS 导航系统相结合,提高了该综合导航系统的准确性和可靠性。该系统利用小波分析算法来减轻图像处理的计算量,提高视觉系统的准确性和解算效率。文献[13]中作者通过实验证了一些以低成本的 GPS 接收机作为小型室外移动机器人导航传感器的可行性。实验的数据是在不同时间不同地点 GPS 接收机所采集的数据,通过对这些数据的分析和比较来进行相关的应用验证。文献[14]的作者利用 GPS 系统信号的反射和折射的统计特性提出了一种改善高大建筑物附近的移动机器人导航信息精度的方法——卡尔曼滤波技术,该技术用来估算 GPS 信号的时变偏移量。文献[15]的作

者通过一系列的实验验证了利用全球导航卫星系统 GNSS 对农业机器人提供精确导航信息的可行性。

随着美国的 SA 政策的取消,虽然民用 GPS 接收机的定位精度得到了大大的提高,但是民用的单频 GPS 接收机的误差依然很大,无法直接应用到移动机器人的导航中。基于实验数据,单频 GPS 接收机的误差在 20 m 左右^[16]。这么大的误差使得民用单频 GPS 接收机不适合单独为移动机器人提供导航信息。并且当机器人在通过高大建筑物附近或周围有高大树木时,由于 GPS 卫星信号的丢失使得 GPS 的定位精度大幅度下降甚至无法正常工作,所以在这种情况下 GPS 也无法持续地为移动机器人提供导航信息。信号的反射和天上卫星不好的相对位置也会对 GPS 的定位精度产生不良的影响^[17]。实验结果表明,当 GPS 接收机所对接的卫星发生改变时,GPS 的精度也会下降。上述所有的因素都表明民用单频 GPS 接收机无法直接单独地为移动机器人提供导航服务,还需要其他导航传感器的辅助。差分 GPS(DGPS)的导航定位精度更准确,可以提供误差小于 1 m 的导航信息。文献[18]中作者提出了一种基于 RTK-DGPS 的室外导航系统。采用这种差分 GPS 接收机使得定位精度可以达到厘米级,但是 DGPS 的价格偏高,并且需要 GPS 基站提供差分信息。基于这些原因,将 GPS 和其他导航传感器的导航信息进行融合以提供连续稳定的、准确的导航服务,这成为了研究的热点。

航迹推算导航(DR)系统是一种经常被用于移动机器人导航定位的导航系统。该系统主要利用编码器、罗盘和陀螺仪等传感器得到机器人当前位置和方向等信息。DR 系统有高精度、快速反应时间和低噪声等优点。然而由于系统误差的存在,使得 DR 系统的误差会随着时间的增加而累积^[19],文献[20]的作者针对 DR 系统的系统误差做了详细的分析。一方面,虽然 DR 系统可以提供短期精确的导航信息,但是由于 DR 系统的误差会随着时间的增加而不断累积,因此不能单独使用其为移动机器人提供长期的导航服务;另一方面,虽然民用 GPS 接收机的精度比较低,反应时间相对慢,而且受到高噪声的影响,但是由于其导航信息不是通过前面导航数据积分而得到的,因此其误差不随着时间的增加而累积。因此将 GPS 和 DR 系统的导航数据进行融合以得到更稳定和准确的导航信息的方法得到了广泛的研究,出现了很多相应的信息融合算法。卡尔曼滤波技术在传感器数据融合中得到了广泛的应用^[21,22]。Chen 等综合了卡尔曼滤波技术与小波神经网络技术,提出了一种改善 GPS/INS 综合导航系统定位精度的信息融合算法,该算法可以在 GPS 无法正常工作时使得导航系统依然可以提供准确的导航信息^[23]。文献[24]中作者提出了一种 GPS 和低精度惯性导航系统的导航信息融合方法。该导航系统信息融合方法基于扩展卡尔曼滤波技术。该系统基于低精度的惯性导航系统和 GPS,应用人工智能思想来对 GPS 和惯性导航系统导航信息进行融合并得到了广泛的研究成果。大部分的人工智能方法考虑的是当前惯性导航系统的误差信

息,而不考虑以前的相应信息。文献[25]中作者利用输入时滞神经网络基于当前和前若干个采样时刻的惯性导航系统的位置和速度信息对惯性导航系统的位置和速度误差进行建模。该结果在 GPS 长时间处于无法使用时可以提供更加可靠的定位结果。文献[26]中作者利用自适应卡尔曼滤波技术来融合惯性测量单元和两个 RTK-GPS 接收机的导航信息,来为移动机器人提供三维的导航定位信息。通过分析,作者提出:综合一个 GPS 和惯性测量单元无法建立起一个可观测系统,而综合两个 GPS 接收机和惯性测量单元可以建立起一个局部可观测系统。这使得由于陀螺仪的漂移而产生的方向估算误差能被弥补。文中还提出了一种自调节滤波器,通过估算 GPS 的测量噪声来处理 GPS 系统无法正常工作时的情况。在 GPS 系统正常工作时,采用卡尔曼滤波技术来进行导航信息的融合。文献[27]中作者为城市中运行的大型机器人设计了一种综合导航系统。采用扩展卡尔曼滤波技术来融合全球导航卫星系统 GNSS 和惯性导航系统的导航信息。文献[28]中作者基于滚动时域卡尔曼有限脉冲响应过滤器提出了一种航迹推算导航系统和 GPS 导航信息融合方法。该滤波技术具有对系统模型误差、初始估算误差和未知偏差等因素鲁棒性强的优点,同时还降低了解算的难度。大部分传统的导航信息融合技术都是基于卡尔曼滤波技术的。卡尔曼滤波方法需要预先知道动态系统的模型。在大多数情况下非线性强的动态系统的建模是一项计算量很大的工作。文献[29]中作者利用人工神经网络系统强大的非线性系统建模的能力来进行 GPS 和航迹推算导航系统的导航信息融合。

卡尔曼滤波技术为导航系统信息融合提供了有力的工具。但是传感器误差模型的不准确、噪声问题和解算量大等问题是基于卡尔曼滤波技术的信息融合算法所需要解决的问题。具体来说卡尔曼滤波技术要求噪声为高斯白噪声,但是由于 GPS 的噪声源非常的多且复杂,因此 GPS 的噪声不可能是高斯白噪声。这使得基于卡尔曼滤波技术的导航信息融合算法的精度下降,在某些情况下甚至使得系统不稳定。解算量大也是这些算法所要面临的一个重要的问题。因此有必要针对民用单频 GPS 接收机的导航数据特点设计相应的导航信息融合算法。

一般来说,移动机器人是在一个已知的环境中工作或利用相应的传感器感知周围的环境然后进行工作的。在工作环境中会存在一些障碍物。一般情况下,移动机器人首先需要确定一条短的、安全的路径。有些时候,移动机器人对路径的要求不仅仅是长度最短,而且还需要考虑转角等其他因素。

移动机器人能够在工作环境中安全顺利地工作的前提是得到一条最优的路径。路径规划的主要任务是找到一条到达目标点的最优路径。随着移动机器人技术的不断发展,路径规划算法得到了广泛的研究^[30~32]。Rashid 等提出了一种可见二叉树算法来为移动机器人进行路径规划^[33]。该算法在考虑机器人与周围障碍物的可见切线的基础上构建所有的可行路径,这些路径组成了一个可见二叉树,

然后进行最短路径的寻找。Duchon 等基于栅格地图改进了著名的 A* 算法,为移动机器人进行路径规划^[34]。改进后的 A* 算法可以在更短的时间内得到最优的路径。Azimirad 和 Shorakaei 提出了一种两极分析进化算法进行最优路径的搜索^[35]。该算法基于间接开环最优控制思想和遗传算法。Dakulović 和 Petrović 为移动机器人路径规划应用提出了一种双向 D* 算法^[36]。该算法基于 Witkowski 规则,可以在二维栅格地图中进行路径优化。文献[37]中作者介绍了一种改进的 A* 算法。改进后的 A* 算法相对于传统的 A* 算法来说需要更少的节点,特别是当环境中存在形状复杂的障碍物时,从而减轻了算法的解算难度。文献[38]中作者基于粒子群优化算法为移动机器人在未知环境中提出了一种多目标路径规划算法。文献[39]中作者基于多神经元启发式搜索算法对传统的 A* 算法进行改进提出了一种在静态环境中进行移动机器人路径优化的算法。文献[40]中作者为多机器人系统提出了一种协同进化改进遗传算法,用于进行路径规划。该改进的遗传算法采用了新的遗传因子,从而改善了传统遗传算法局部最优解的问题,并且加快了系统的收敛速度。文献[41]中作者基于生态地理学理论提出了一种改进的粒子群优化算法。该算法可以在静态环境中为移动机器人提供全局最优路径。文献[42]中作者基于人工势场方法和细菌进化算法在动态环境中为移动机器人提供最优的路径。Espelosín 等基于粒子群算法考虑了在二维动态环境中进行移动机器人路径优化的问题^[43]。Huang 等基于细胞自动机和二型模糊逻辑提出了一种路径规划算法^[44]。文献[45]中作者提出了一种移动机器人路径规划的算法。首先基于人工蜂群算法进行局部的搜索,然后利用进化规划算法进行最优路径的提炼。文献[46]中作者在三维环境中基于 A* 算法提出了一种移动机器人路径规划算法。

当前,常用的路径规划方法有人工势场法^[47],模糊逻辑法^[48],A* 算法和神经网络法^[49]等。人工势场法可以提供一个安全的路径,但是该方法容易陷入局部最优解。在某些层面上来说,模糊逻辑法具有一定的智能性。模糊逻辑法的问题在于其缺乏自适应性,该方法仅对某类问题可以提供好的结果。A* 算法可以提供好的路径规划结果,但是在大的搜索环境下其效率比较低。A* 算法利用离线和在线学习神经网络具备了很强的学习能力和自适应性,但是如何得到最优的神经网络结构仍是个难题。而且如何训练神经网络也是一个必须解决的难题。

近些年来应用群体智能来解决路径规划问题的研究越来越多^[50~54]。蚁群算法经常被用于解决包括移动机器人路径规划在内的各种优化问题^[55~57],但是蚁群算法有容易陷入局部最优解和收敛速度慢等问题。虽然针对这些问题已经有了很多的研究结果,但是大部分仅是针对其中的一个问题取得的成果。

在得到了最优路径后,基于导航系统提供的导航信息,移动机器人完成任务时,通常需要跟踪事先设定好的路径。轮式移动机器人控制的两个主要问题是:如

何进行轨迹跟踪控制和点位控制。虽然点位控制问题在理论上很难解决,但是在实践中的应用并不广泛^[58]。轮式移动机器人的控制算法通常是在一个预先设定好的无障碍的路径的基础上工作的。因此,在实际上,轨迹跟踪控制问题得到了广泛关注。移动机器人的轨迹跟踪控制问题是控制机器人跟踪一个给定的、时变的轨迹。一般来说,其目的是使机器人能够在每一个采样周期上以特定的姿态到达预定的位置。

由于控制系统的设计难度和广泛的应用领域,轮式移动机器人路径跟踪控制问题得到了很多的关注^[59~61]。由于大多数的任务涉及跟踪一个事先设计好的安全路径,因此强大的路径跟踪能力对于移动机器人来说非常重要。

轮式移动机器人路径跟踪控制要比点位控制简单得多,这是因为在设计路径跟踪控制算法时假设轮式移动机器人的轮子与地面良好接触,运动时无滑动。这使得轮式移动机器人成为了一种非完整控制系统。轮式移动机器人路径跟踪控制问题已经成为了机器人领域的研究热点之一^[62~67]。文献[68,69]中作者提出了一种线性跟踪控制器。然而,该控制算法仅当机器人的线性速度不为零时才管用。这样当要控制机器人跟踪某一设计好的路径同时在某一特定点上停下来时,就很困难了。因此更多的非线性控制器被应用到该领域当中^[70~73]。Zenon 和 Marcin 的基于自校正设计,为轮式移动机器人提出了一种跟踪控制算法^[74]。Lim 等为非完整轮式移动机器人提出了一种非线性模型预测跟踪控制算法^[75]。Dinh 等为移动机器人提出了一种微分滑膜跟踪控制算法^[76]。在以上大部分的研究中是基于无扰动且轮子与地面无滑动的假设的。然而,实际上由于轮式移动机器人的速度和载重不同,有些情况下轮子与地面产生滑动是不可避免的,因此也有不少研究考虑了这种情况^[77,78]。

第2章 GPS/DR 室外移动机器人导航系统

机器人是靠自身动力和控制能力来实现各种功能的一种机器^[79]。它既可以接受人类指挥,又可以运行预先编排的程序,也可以根据人工智能技术制订的原则纲领行动。它的任务是协助或取代人类完成工作,例如,在生产业、建筑业,或是其他有危险性的行业进行工作。机器人技术的发展主要有三个阶段。第一阶段为简单个体机器人,属于示教再现型。它只能死板地按照人给它规定的程序工作,不管外界条件有何变化,自己都不能对所做的工作做出相应的调整。如果要改变机器人所做的工作,必须由人对程序做相应的改变,因此它是毫无智能的。第二阶段为群体劳动机器人,对外界信息能够反馈,有一定的触觉、视觉、听觉。第三阶段为类人智能机器人,它不仅具有感知和理解周围环境的能力,而且还具有独立判断和行动的能力,并具有记忆、推理和决策的能力,因而能够完成更加复杂的动作。自主移动机器人属于类人智能机器人,是本体自带各种必要的传感器、控制器,在运行过程中无外界人为信息输入和控制的条件下,可以独立完成一定任务的机器人。它集环境感知、动态决策与规划、行为控制与执行等功能于一体,具有高度的自动化程度。其自主性主要体现在对环境的感知和理解、行为自主规划,以及自我学习和自我适应等能力方面。作为一种复杂的智能系统,自主移动机器人的研究涉及计算机视觉、模式识别、传感器及多传感器感知与信息融合技术、人工智能、自动控制等诸多学科的理论和技术,集中体现了计算机技术和人工智能的最新成果,在军事和民用方面显示出越来越广泛的应用价值。

2.1 常用的机器人导航系统

近几年,移动机器人技术得到飞速发展。随着它的应用领域越来越广泛,机器人被期望应用在偏远、危险的室外环境。特别是在变电站、化工厂等高温、高辐射和强干扰的场所中,代替人类进行全天候的作业。作为机器人在户外作业的基础,定位成了需要解决的首要问题。机器人需要精确地知道自身当前的位置,才能完成局部避障或是全局规划,这是实现其智能化的关键。所谓定位是指在结构化或非结构化环境中,机器人利用先验环境信息、机器人位姿的当前估计及传感器的观测值等输入信息,经过一定的处理和变换,产生更加准确的对机器人当前位姿的估计^[80]。根据定位方式和传感器的不同,一般可以把定位技术分为以下几个大类,即信号灯定位、基于地图的定位、路标定位、基于视觉的定位、航迹推算定位、无线

电定位、惯性导航、声学导航、卫星导航和综合导航等。

激光传感器是通过测距来定位的,激光传感器发射一个激光信号,根据收到从物体反射回来的信号的时间差来计算这段距离,根据发射激光的角度来确定物体和发射器的角度,从而得出物体与发射器的相对位置。近年来,激光传感器在移动机器人定位导航中的应用越来越广泛,这主要是因为激光传感器在距离测量方面具有高频率、高精度等优点。激光传感器所发射的扫描光束能够快速、大量、准确地获取周围环境的距离信息。此外,激光雷达是利用自主发射激光束的方式获取环境信息的,因此,不受环境光照变化、电磁干扰等因素的影响。然而,目前常用的激光传感器扫描的分辨率较低,在动态环境中噪声较大,在室外环境中的定位仍存在不足。

基于无线局域网的定位系统,是在一定的区域内安装适量的无线基站,根据这些基站获得的待定位物体发送的信息(如时间和信号强度等),并结合基站所组成的拓扑结构综合分析,从而确定物体的具体位置^[81]。这类系统可以利用现有的无线局域网设备,仅需要增加相应的信息分析服务器以完成定位信息的分析。无线局域网的网络速度与以太网相当,传输范围可达到几十千米,并且具有移动性高、保密性好的优点,因此它在机器人定位上也开始逐步使用。然而无线传感器网络较为密集,并且易受电磁干扰,节点的计算、存储和通信等能力也有限,这限制了其在室外定位中,尤其是在具有强电磁干扰的特殊环境中的应用。

视觉定位可以描述为运动载体通过视觉设备观察场景,再通过图像分析、目标识别等技术,计算载体在世界坐标系下的全局位姿,或是载体相对场景中特定参照物的局部相对位姿^[82]。目前基于视觉定位的室外移动机器人自定位的关键技术主要体现在路标选择、视觉系统选择、图像理解和分析、定位精度及实时性要求等方面。路标可分为两种:一种是自然路标,是直接从环境中提取的已存在的特征及物体;另一种是人工路标,是为了实现机器人导航和定位而人为放置在机器人工作环境中的标志物体。自然路标运用灵活,适合于需要高速运动的机器人(如智能车辆),但对路况要求严格,而且识别难度大,难以使用。人工路标适合于速度慢但对精度要求高的作业型机器人,它可以通过人为地设计和摆放路标,让机器人能够鲁棒、快速地识别出路标进行定位,因而得到了广泛的应用。图像理解和分析是整个视觉定位算法的核心。在室外环境下获取目标图像后,需要从中识别出用于定位的路标物。其难点在于如何选择合适的特征或结构代表图像信息,并对诸多环境不确定因素有较好的鲁棒性和不变性,同时满足实时性要求。目前常用的识别方法包括图像整体匹配和局部特征点匹配。视觉定位指标通常包括定位精度、实时性、工作环境、定位设备、应用场合等方面。根据不同的应用,需要使用不同的指标进行衡量,一般定位精度和实时性是两个主要的考核指标。定位精度是指用算法计算结果与实际位姿的偏差进行衡量,包括坐标误差和角度误差,受成像设备分辨

率和算法性能影响。实时性是算法处理一帧图像的时间反映,通常实时算法要求每秒处理30幅或以上图像。

惯性导航系统(INS)是一种不依赖于外部信息、也不向外部辐射能量的自主式导航系统,其工作环境不仅包括空中、地面,还包括水面。惯性导航系统的基本工作原理是以牛顿力学定律为基础,通过测量载体在惯性参考系中的加速度,将它对时间进行积分,且把它变换到导航坐标系中,得到在导航坐标系中的速度、偏航角和位置等信息。

惯性导航属于推算导航的一种。即从一已知点的位置根据连续测得的运载体航向角和速度推算出其下一点的位置,因而可连续测出运动体的当前位置。惯性导航系统中的陀螺仪用来形成一个导航坐标系使加速度计的测量轴稳定在该坐标系中并给出航向和姿态角;加速度计用来测量运动体的加速度,经过对时间的一次积分得到速度,速度再经过对时间的一次积分即可得到距离。

惯性导航系统有如下优点。(1)由于它不依赖于任何外部信息,也不向外部辐射能量的自主式系统,故隐蔽性好,不受外界电磁干扰的影响。(2)可全天候、全球、全时间地工作于空中、地球表面乃至水面上。(3)能提供位置、速度、航向和姿态角数据,所产生的导航信息连续性好而且噪声低。(4)数据更新率高、短期精度和稳定性好。

惯性导航系统有如下缺点。(1)由于导航信息经过积分而产生,定位误差随时间而增大,长期精度差。也就是说惯性导航系统无法长期的单独提供精确的导航信息,其误差随着时间的增加而增加,而且是没有上限的。(2)每次使用之前需要较长的初始对准时间。(3)设备的价格较昂贵。(4)不能给出时间信息。

惯性导航系统目前已经发展出挠性惯导、光纤惯导、激光惯导、微固态惯性仪表等多种方式。陀螺仪由传统的绕线陀螺发展到静电陀螺、激光陀螺、光纤陀螺、微机械陀螺等。激光陀螺测量动态范围宽,线性度好,性能稳定,具有良好的温度稳定性和重复性,在高精度的应用领域中一直占据着主导位置。由于科技进步,成本较低的光纤陀螺(FOG)和微机械陀螺(MEMS)精度越来越高,是未来陀螺技术发展的方向。我国的惯性导航技术近年来已经取得了长足进步,液浮陀螺平台惯性导航系统、动力调谐陀螺四轴平台系统已相继应用于长征系列运载火箭中。其他各类小型化捷联惯导、光纤陀螺惯导、激光陀螺惯导及匹配GPS修正的惯导装置等也已经大量应用于战术制导武器、飞机、舰艇、运载火箭、宇宙飞船等领域。如漂移率 $0.01^{\circ}/h$ ~ $0.02^{\circ}/h$ 的新型激光陀螺捷联系统在新型战机上试飞,漂移率 $0.05^{\circ}/h$ 以下的光纤陀螺、捷联惯导在舰艇、潜艇上的应用,以及小型化挠性捷联惯导在各类导弹制导武器上的应用。

GPS(global positioning system)是一种最常用的卫星导航系统。GPS最初只是应用到军事方面的导航系统,现在已经广泛地民用了。GPS导航系统需要天空

中至少要有 24 颗 GPS 卫星在工作。每颗卫星都连续地发送包含其当前位置和当前 GPS 时间数据的信息。每颗 GPS 通信卫星都是同步发射信号的,当一个 GPS 接收机接收到某个卫星的信号后,根据当前的时间和信号发送时的时间差就可以求出 GPS 接收机与该卫星之间的距离。GPS 接收机就是利用这个得到的伪距来计算自己的位置的。当一个 GPS 接收机同时接收到 4 个或更多的卫星的信号时,就可以计算出其当前的位置,请参看图 2-1。

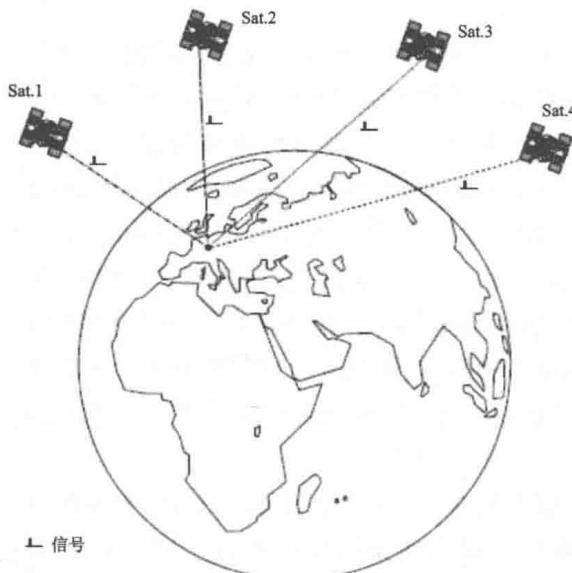


图 2-1 至少需要 4 颗卫星来进行三维定位

GPS、Galileo 和 Glonass 系统都是常用的卫星导航系统。GPS 以其定位精度高和覆盖范围广而得到了广泛的应用。在室外移动机器人导航中, GPS 接收机有时候会丢失卫星的信号,这就导致导航精度的降低,甚至会失去导航系统的可靠性。GPS、Galileo 和 Glonass 系统的结合使用可以使天空中可用的导航卫星得到大幅度的提高,从而提高移动机器人导航的可靠性和精度。而且通过结合移动机器人自身的运动性能,该系统可以进一步提高室外移动机器人导航的精度和可靠性。近些年来,虽然美国取消了 SA 政策,但是民用 GPS 的精度依然很低。所以关于提高 GPS 导航定位精度的算法得到了广泛的关注。很多学者提出了通过改善 GPS 数据解算的方法来提高 GPS 的定位精度。例如建立各种各样的 GPS 的误差模型来提高 GPS 的定位精度,采用最小二乘法循环推算的方法来提高 GPS 信息解算的速度和精度,以及利用 GPS 的载波相位与伪距定位相结合的方法来提高 GPS 的定位精度等。

差分 GPS 也称为 DGPS,它可以将 GPS 的公共误差抵消,例如卫星时钟偏差、星历误差、电离层误差、对流层误差等,从而提高了定位精度。DGPS 的定位精度

很高,其定位误差可以达到 1 m,甚至更低。因此基于 DGPS 高定位精度的特点,很多学者提出采用 DGPS 对移动机器人进行导航。DGPS 有定位精度高、使用方便、可以全天候实时提供导航定位信息等优点,但是其价格非常昂贵,并且 DGPS 需要一个基站来为其提供差分信息。DGPS 的定位精度随着 GPS 接收机与基站的距离的改变而改变,而且当 GPS 接收机无法收到基站的差分信息时,DGPS 的精度会大幅度下降。更为重要的是,卫星信号的可用性取决于能否在一段时间内获得稳定的、从相同卫星发出信号的能力。在移动导航中,GPS 接收机定位精度通常受到卫星信号状况和道路环境的影响,同时还受到诸如时钟误差、传播误差、接收机噪声等诸多因素的影响,接收到的卫星信号可能是突发的、断续的。因此,单独使用 DGPS 无法长期稳定地为移动机器人提供准确的导航信息。有学者提出使用两个单频 GPS 接收机来实现 DGPS 的功能,从而节省了购买 DGPS 的花销,但是这需要两个 GPS 接收机能够同时使用相同的 GPS 卫星进行导航定位,在很多情况下无法满足这个条件。

综合导航系统是综合利用多种导航传感器,通过信息融合技术将来自多个传感器的导航信息进行融合,从而得到比任何一个单独传感器更精确和稳定的导航结果。综合导航系统的关键是导航信息融合技术。信息融合技术是近年来十分热门的研究课题,它结合了控制理论、信号处理、人工智能、概率和统计的发展。信息融合是指综合来自多个传感器的感知数据,以产生更可靠、更准确或更精确的信息。经过融合的多传感器系统能完善地、精确地反映检测对象特性,消除信息的不确定性,提高传感器的可靠性。经过融合的多传感器信息具有以下特性:信息的冗余性、信息的互补性、信息的实时性和信息的低成本性。对于 GPS 和航迹推算导航系统来说,卡尔曼滤波技术和扩展卡尔曼滤波技术是最常用的信息融合算法。但是卡尔曼滤波有其自身的局限性。卡尔曼滤波基于高斯白噪声的假设不符合 GPS 误差源的特点,使得卡尔曼滤波的效果不好,甚至会导致融合结果发散,从而使得整个导航系统无法正常工作。并且, GPS 接收机存在无法接收到足够数目的导航卫星的情况。在这种情况下 GPS 将无法正常提供导航信息。卡尔曼滤波技术无法解决当 GPS 无法正常使用时导航系统的问题。

本书的一个主要内容是对室外移动机器人的导航系统的研究。航迹推算法是常用的移动机器人的导航方法。这种方法可以提供短期的非常准确的导航结果,但是由于导航信息经过积分而产生,定位误差随时间的增大而增大,长期精度差。也就是说航迹推算导航系统无法长期地单独提供精确的导航信息,其误差随着时间的增加而增加,而且是没有上限的。GPS 导航系统是最常用的全球卫星导航系统之一,它可以提供准确的全天候的导航信息而不受时间空间和天气条件等的限制。虽然 SA 政策已经被取消,但是民用的 GPS 接收机的误差还是比较大的。DGPS 系统是可以提供非常精确的导航定位结果的,但是 DGPS 的价格非常昂贵,

而且其需要一个基准站来提供差分的信息。其精确度随着导航 GPS 接收机与基准站的距离的改变而改变,而且在某些时候 GPS 接收机无法接收到基准站的信息时其导航误差将会大幅度的增大。简单便宜的单频 GPS 接收机的单点定位的绝对误差很大,但是其误差是有一定范围的。而航迹推算法可以提供短期的非常精确的导航信息,但是其误差会随着时间增大而无限制的积累。所以,利用航迹推算导航系统和单频 GPS 接收机的互补的特点,提出了一套适合室外移动机器人导航的综合导航系统。此方法利用航迹推算法提供短期的准确的导航结果,同时利用单频 GPS 接收机的导航信息来对长期的导航结果进行修正,从而得出一个准确的、稳定的、长期可用的导航方法。

2.2 GPS 定位导航

要设计出适合于融合航迹推算导航系统和民用单频 GPS 系统导航数据的信息融合算法,该算法需要基于所选择的单频 GPS 接收机的数据特点。因此,首先要熟悉 GPS 的定位原理和影响误差的主要因素。

1. GPS 系统的定位原理和影响误差的主要因素

GPS 系统可以为军用和民用导航提供定位信息。其标准定位服务信号可以为民用导航提供服务,而其精确定位服务信号最早只能为美国军方提供导航定位服务。第一颗 GPS 导航卫星于 1978 年 2 月升空。现今已经有多颗 GPS 导航卫星在太空中提供导航服务。这些 GPS 导航卫星分布在距地面 20180 km 的 6 个不同的空间轨道平面上。GPS 系统由三个部分组成:①空间部分,包括所有的导航卫星;②控制部分,包括地面站的监视系统、主控制站等;③用户部分,包括所有的军用和民用 GPS 用户。

利用 GPS 系统可以在任何时间任何地点得到准确的定位信息(包括经度、纬度和高度),同时能够得到准确的时间信息。而且可以对得到的信息进行计算,从而得到准确的速度和行驶方向的信息。这些信息数据来自围绕地球转动的 28 颗 GPS 定位卫星。GPS 定位卫星同步发送的数据包括:①卫星时钟时间和同步信号;②精确的轨道数据;③确定准确的卫星时钟时间的时间矫正信息;④所有卫星近似的轨道信息;⑤计算信号传送时间的校正信号;⑥电离层的数据;⑦导航卫星健康情况的数据。GPS 接收机接收到来自 GPS 定位卫星的信号后,根据信号中信号发送时间可以得到卫星与 GPS 接收机之间的距离,同时可获得该卫星在天空中的位置。

一般来说,求三维坐标值只是需要 3 颗卫星与 GPS 接收机的距离就可以了。但是实际中,由于 GPS 接收机的时间和 GPS 卫星的时间有误差,所以需要知道至少 4 个卫星与 GPS 接收机的距离来确定 GPS 接收机的三维坐标值和这个时间的

差值。设 R_i ($i=1,2,3,4$) 为卫星 i 距离 GPS 接收机的实际距离, c 是光的速度, Δt_i 是信号从卫星 i 传到 GPS 接收机的实际时间, Δt_{0i} 是卫星 i 与 GPS 接收机的时间差, PSR_i 是卫星 i 距离接收机的伪距。令 X_{s1}, Y_{s1}, Z_{s1} 为卫星 i 的三维坐标, X_u, Y_u, Z_u 是 GPS 接收机的三维坐标, 则

$$\begin{cases} \text{PSR}_1 = R_1 + c\Delta t_{01} \\ \text{PSR}_2 = R_2 + c\Delta t_{02} \\ \text{PSR}_3 = R_3 + c\Delta t_{03} \\ \text{PSR}_4 = R_4 + c\Delta t_{04} \end{cases} \quad (2-1)$$

$$\begin{cases} R_1 = \sqrt{(X_{s1} - X_u)^2 + (Y_{s1} - Y_u)^2 + (Z_{s1} - Z_u)^2} \\ R_2 = \sqrt{(X_{s2} - X_u)^2 + (Y_{s2} - Y_u)^2 + (Z_{s2} - Z_u)^2} \\ R_3 = \sqrt{(X_{s3} - X_u)^2 + (Y_{s3} - Y_u)^2 + (Z_{s3} - Z_u)^2} \\ R_4 = \sqrt{(X_{s4} - X_u)^2 + (Y_{s4} - Y_u)^2 + (Z_{s4} - Z_u)^2} \end{cases} \quad (2-2)$$

将公式(2-2), 代入公式(2-1)得

$$\begin{cases} \text{PSR}_1 = \sqrt{(X_{s1} - X_u)^2 + (Y_{s1} - Y_u)^2 + (Z_{s1} - Z_u)^2} + c\Delta t_{01} \\ \text{PSR}_2 = \sqrt{(X_{s2} - X_u)^2 + (Y_{s2} - Y_u)^2 + (Z_{s2} - Z_u)^2} + c\Delta t_{02} \\ \text{PSR}_3 = \sqrt{(X_{s3} - X_u)^2 + (Y_{s3} - Y_u)^2 + (Z_{s3} - Z_u)^2} + c\Delta t_{03} \\ \text{PSR}_4 = \sqrt{(X_{s4} - X_u)^2 + (Y_{s4} - Y_u)^2 + (Z_{s4} - Z_u)^2} + c\Delta t_{04} \end{cases} \quad (2-3)$$

我们可以通过求解方程(2-3)来得到 GPS 接收机的三维坐标。

在方程(2-3)的求解过程中, 没有考虑各种误差干扰因素。对于 GPS 导航系统来说, 其主要的误差源有以下几种。

(1) 卫星时钟误差。虽然每个 GPS 导航卫星上都配有精确的原子钟, 但是依然存在着微小的卫星时钟误差。一个 10 ns 的误差会导致定位的 3 m 误差。

(2) 卫星轨道误差。一般来说卫星轨道误差在 1~5 m。

(3) 光速产生的误差。从 GPS 导航卫星中发射的信号以光速的速度发送到用户接收机。在计算中我们近似认为光速是恒定的, 但是当信号进入电离层和对流层时, 速度会减慢, 无法维持恒定的速度, 因此会导致计算的误差。

(4) 多路径效应。GPS 导航卫星发射的信号由于周围高大建筑物等的反射, 使得 GPS 接收机接收到的信号有直接从 GPS 导航卫星发射的信号, 也有反射回来的信号, 产生了多路径效应, 会影响 GPS 接收机对位置的解算。

(5) GPS 导航卫星在天空中的位置。如果用来定位的四颗 GPS 导航卫星彼此间离得太近, 定位的精度会下降。GPS 导航卫星的位置对定位精度的影响用参数 GDOP 来表示。GDOP 的值越小, 表示当前的 GPS 导航卫星的相对位置越好, 对定位精度的影响越小; 反之, GDOP 的值越大, 则当前的 GPS 导航卫星的相对位

置对定位精度的影响越大。

2. GPS 系统定位实验

普通民用 GPS 接收机价格适中,但是定位精度差,无法单独提供定位导航服务。本书的内容是将航迹推算导航系统和 GPS 系统相融合,从而得到更为准确和稳定的导航信息。这就需要一个有效的信息融合算法,该信息融合算法应基于 GPS 数据误差的特点。所以有必要对所选择的 GPS 接收机进行大量的实验,并对实验数据进行整理和分析,以期得出该 GPS 接收机的数据误差特点。

首先是 GPS 接收机的选择。所选的 GPS 接收机要求价格便宜,以达到设计算法的初衷。经过挑选,我们选择了两款 GPS 接收机。第一款的型号是 Ascen GPS 730,如图 2-2 所示;另一款是更为便宜的普通单频 GPS 接收机,如图 2-3 所示。

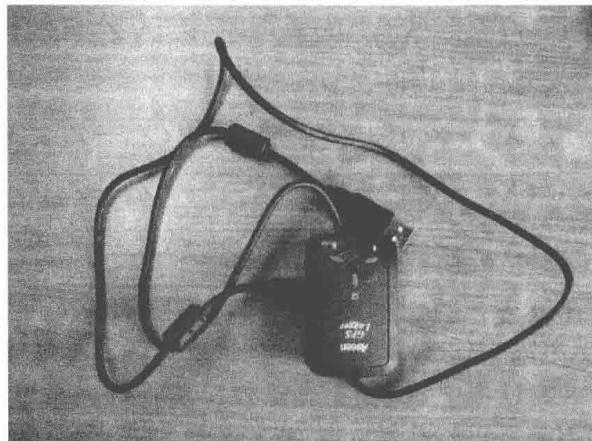


图 2-2 Ascen GPS 730 接收机

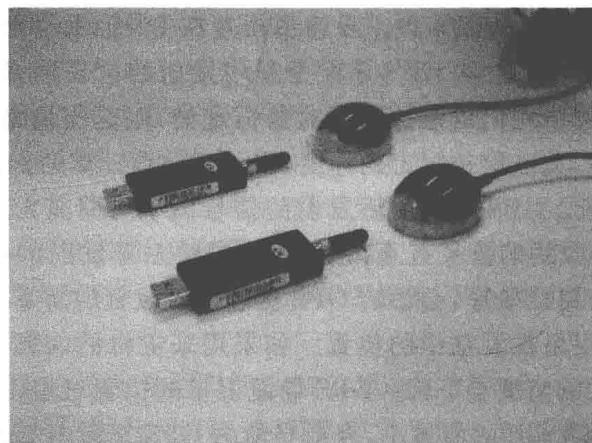


图 2-3 普通单频 GPS 接收机

在进行实验之前,首先针对 GPS 接收机的数据特点,用 VC 编制 GPS 数据信息的收集程序和定位数据整理程序。GPS 接收机解算后给出的信息如下。

推荐定位信息(GPRMC)。数据格式为:“\$GPRMC,〈1〉,〈2〉,〈3〉,〈4〉,〈5〉,〈6〉,〈7〉,〈8〉,〈9〉,〈10〉,〈11〉,〈12〉*hh”。

〈1〉:UTC 时间, hhmmss(时分秒)格式。

〈2〉:定位状态, A=有效定位, V=无效定位。

〈3〉:纬度 ddmm. mmmm(度分)格式(前面的 0 也将被传输)。

〈4〉:纬度半球 N(北半球)或 S(南半球)。

〈5〉:经度 dddmm. mmmm(度分)格式(前面的 0 也将被传输)。

〈6〉:经度半球 E(东经)或 W(西经)。

〈7〉:地面速率(000.0~999.9 节,前面的 0 也将被传输)。

〈8〉:地面航向(000.0~359.9°,以正北为参考基准,前面的 0 也将被传输)。

〈9〉:UTC 日期, ddmmyy(日月年)格式。

〈10〉:磁偏角(000.0~180.0°,前面的 0 也将被传输)。

〈11〉:磁偏角方向, E(东)或 W(西)。

〈12〉:模式指示(仅 NMEA0183 3.00 版本输出, A=自主定位, D=差分, E=估算, N=数据无效)。

当前卫星信息(\$GPGSA)。数据格式为:“\$GPGSA,〈1〉,〈2〉,〈3〉,〈3〉,〈3〉,〈3〉,〈4〉,〈5〉,〈6〉,〈7〉”。

〈1〉:模式:M=手动, A=自动。

〈2〉:定位形式:1=未定位, 2=二维定位, 3=三维定位。

〈3〉:PRN 数字:01 至 32 表天空使用中的卫星编号, 最多可接收 12 颗卫星信息。

〈4〉:PDOP 位置精度因子(0.5~99.9)。

〈5〉:HDOP 水平精度因子(0.5~99.9)。

〈6〉:VDOP 垂直精度因子(0.5~99.9)。

〈7〉:Checksum. (检查位)。

可视卫星状态输出语句(\$GPGSV)。数据格式为: \$GPGSV,(1),(2),(3),(4),(5),(6),(7),…,(4),(5),(6),(7)*hh(CR)(LF)。

(1):总的 GSV 语句电文数。

(2):当前 GSV 语句号。

(3):可视卫星总数。

(4): 卫星号。

(5): 仰角(00~90°)。

(6): 方位角(000~359°)。

(7): 信噪比(00~99 dB)。

*: 总和校验域。

hh: 总和校验数。

(CR)(LF): 回车, 换行。

其中, 每条语句最多包括四颗卫星的信息, 每颗卫星的信息有四个数据项, 即(4)卫星号, (5)仰角, (6)方位角, (7)信噪比。

地面速度信息(GPVTG)。数据格式为: \$ GPVTG, <1>, T, <2>, M, <3>, N, <4>, K, <5> * hh。

<1>: 以真北为参考基准的地面航向(000~359°, 前面的 0 也将被传输)。

<2>: 以磁北为参考基准的地面航向(000~359°, 前面的 0 也将被传输)。

<3>: 地面速率(000.0~999.9 节, 前面的 0 也将被传输)。

<4>: 地面速率(0000.0~1851.8 km/h, 前面的 0 也将被传输)。

<5>: 模式指示(仅 NMEA0183 3.00 版本输出, A 表示自主定位, D 表示差分, E 表示估算, N 表示数据无效)。

GPS 固定数据输出语句(\$GPGGA)。这是一帧 GPS 定位的主要数据, 也是使用最广的数据。\$GPGGA 语句包括 17 个字段: 语句标识头, 世界时间, 纬度, 纬度半球, 经度, 经度半球, 定位质量指示, 使用卫星数量, 水平精确度, 海拔高度, 高度单位, 大地水准面高度, 高度单位, DGPS 数据期限, 差分参考基站标号, 校验和结束标记(分别用回车符(CR)和换行符(LF)), 分别用 14 个逗号进行分隔。该数据帧的结构及各字段释义如下。

\$GPGGA: 起始引导符及语句格式说明(本句为 GPS 定位数据)。

<1>: UTC 时间, 格式为 hhmmss.sss。

<2>: 纬度, 格式为 ddmm. mmmm(第一位是零也将传送)。

<3>: 纬度半球, N 或 S(北纬或南纬)。

<4>: 经度, 格式为 dddmm. mmmm(第一位是零也将传送)。

<5>: 经度半球, E 或 W(东经或西经)。

<6>: 定位质量指示, 0=定位无效, 1=定位有效。

<7>: 使用卫星数量, 从 00 到 12(第一位是零也将传送)。

<8>: 水平精确度, 0.5 到 99.9。

〈9〉:天线离海平面的高度, -9999.9~9999.9 m。

〈10〉:大地水准面高度, -9999.9~9999.9 m。

〈11〉:DGPS 数据期限(RTCM SC-104),最后设立 RTCM 传送的秒数量。

〈12〉:差分参考基站标号,从 0000 到 1023(首位 0 也将传送)。

* 语句结束标志符。

xx 从 \$ 开始到 * 之间的所有 ASCII 码的异或校验和。

GPS 接收机通过 RS232 口与电脑相连。GPS 的数据通过 RS232 口以串行通信的形式发送到电脑里面。GPS 数据的更新率为 1 Hz, 即每秒 GPS 接收机发送一次定位数据。首先, 我们用 VC 编制收集 GPS 接收机的定位数据平台。利用该数据收集平台, 可以将每秒钟接收到的 GPS 定位数据写入一个文本文件中。在这么多的 GPS 数据里面, 我们主要感兴趣的是 GPS 固定数据输出语句(\$GPGGA)中的 UTC 时间(格式为 hhmmss. sss), 纬度(格式为 ddmm. mmmm), 纬度半球(N 或 S), 经度(格式为 dddmm. mmmm), 经度半球(E 或 W), 定位质量指示(0 表示定位无效, 1 表示定位有效), 使用卫星数量(从 00 到 12), 天线离海平面的高度和推荐定位信息(GPRMC)中的 PDOP 位置精度因子。在实验完毕后, 所有实验期间得到的 GPS 定位数据被写入一个文本文件中。我们利用 VC 编制读取数据程序, 在实验中写入定位数据的文本文件中, 将我们想要的数据读出来再写到另一个文本文件中。然后用 Matlab 进行数据的初步处理, 将数据中的定位质量指示为无效的定位数据和 PDOP 位置精度因子过大的定位数据剔除。PDOP 位置精度因子的大小决定了 GPS 接收机探测到的用来定位的 GPS 导航卫星在天空中的几何位置对 GPS 定位精度的影响。如果用于定位的 GPS 导航卫星的位置彼此距离太近, 则会增加 GPS 系统定位的误差, 这在 PDOP 中反映为 PDOP 的值偏大。因此, 在数据中应该剔除 PDOP 偏大的数据。

实验的第一步, 我们先找到了一个已知精确坐标点的测量点。在此测量点上分别用所选用的两款 GPS 接收机进行 GPS 定位实验。在实验中我们发现, 有些时候 GPS 的定位数据会突然发生变化, 定位精度突然变得很低。经过多次的实验和分析, 发现 GPS 数据的突变原因大致有两点。其一, 由于天气和时间等变化, 导致 GPS 接收机所探测到的 GPS 导航卫星发生了变化, 卫星数目相应减少, 从而导致 GPS 定位数据的突变, 这个是自然的因素。其二, 当用于采集数据的笔记本电脑的电量偏低时, 也会导致 GPS 接收机所接收到的信号偏弱, 从而使得 GPS 定位精度下降。因此, 我们更换了原来的小型笔记本电脑, 换成了电池电量充足的大笔记本电脑, 从而使得接收到的 GPS 定位信息更加稳定。在这个已知精确坐标的测

量点上,我们用所选择的 GPS 接收机进行定位数据收集。对于这个精确知道坐标的测量点,我们选择进行测量的条件是:天气情况良好,晴天、少云或无云,时间选择上午 9 点左右或下午 3 点左右。在这样的条件下, GPS 接收机所能探测到的 GPS 导航卫星数比较多,因此其定位精度比较高。在每次测量实验中,我们测量的时间为 40~50 min。得到测量数据后,我们通常剔除前 3~5 min 得到的数据。这样可以利用充分稳定后的 GPS 定位数据来进行整理和分析。经过多天多次的测量,我们对在该点上得到的定位数据进行整理和分析。图 2-4 所示的是利用第一款所选 GPS 接收机进行测量时的结果。经过对测量数据的整理,我们得到该 GPS 接收机的绝对定位误差为 4~6 m,随机误差为 1~3 m。这款 GPS 接收机的定位误差相对来说比较低,其随机误差相对较小。图 2-5 所示的是利用第二款所选 GPS 接收机进行测量时的结果。这张图是经过数据处理过的图,其上的数据是用所得到的数据减去其平均值后得到的数据。经过对测量数据的整理我们得到该 GPS 接收机的绝对定位误差为 12~15 m,随机误差为 5~10 m。这款 GPS 接收机的定位误差相对来说比较大,其随机误差相对较大。

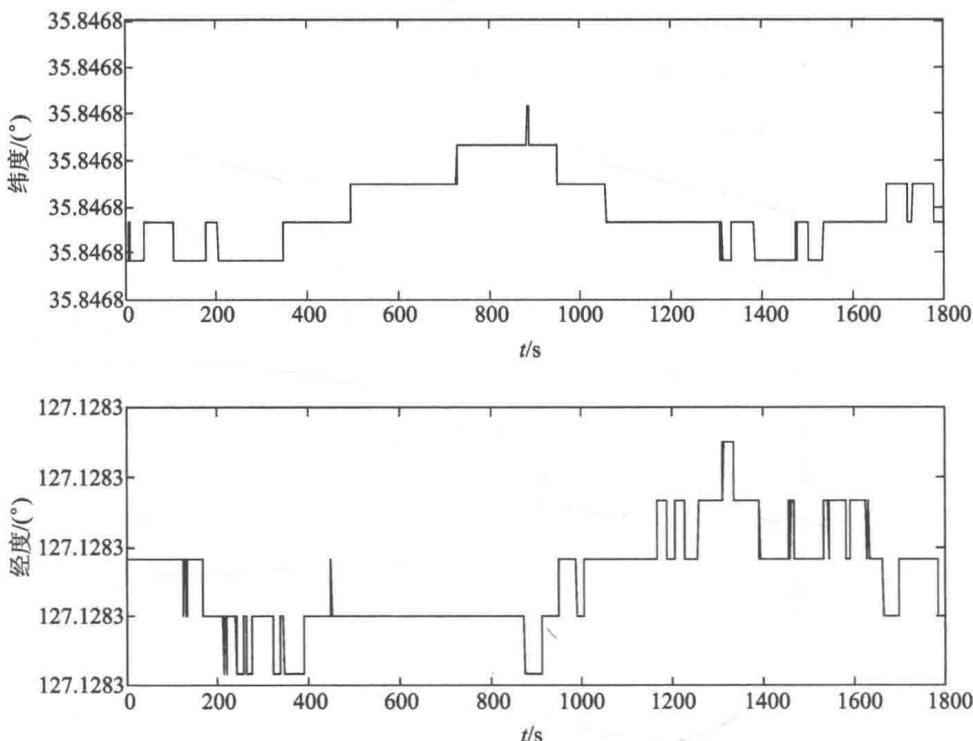


图 2-4 第一款 GPS 接收机的测量数据

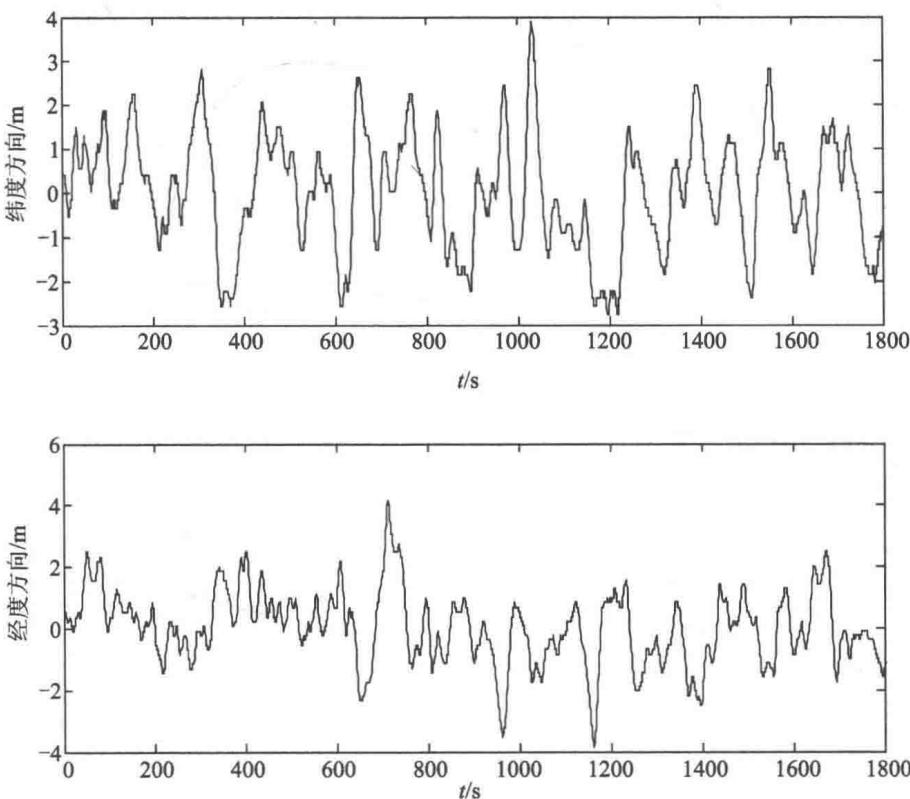


图 2-5 第二款 GPS 接收机的测量数据

实验的第二步,选取若干个有代表性的测量点。这些测量点包括在水面上的测量点、空旷地带的测量点、周围有灌木丛的测量点、周围有高大树木的测量点,以及周围有高大建筑物的测量点等。在这些测量点上,我们分别在凌晨、清晨、上午、中午、下午、晚间、半夜等不同的时间段,在晴天、多云、阴天等不同的天气情况下多次进行 GPS 测量实验。对数据进行整理后,我们发现,不同的时间、地点和天气情况下所得到的测量数据精度也不同。在空旷地带的测量点得到的数据精度较高,水面上的测量点由于水面的反射使得多路径效应的影响比较大,降低了测量精度,周围有树木或高大建筑物的测量点由于多路径效应和所探测 GPS 定位卫星数目减少使得定位精度下降。由于 GPS 导航卫星发射的定位信号很弱,不同的天气也会导致 GPS 接收机探测 GPS 卫星数目不同和探测信号强度不同而导致定位精度不同。另外不同的时间段会由于探测到的 GPS 定位卫星的数目和位置不同使得定位精度不同。经过 GPS 定位数据的整理和分析,我们发现,虽然在不同的环境、天气和时间上这两款 GPS 接收机的定位精度有所不同,但是其得到的数据的走向趋势大致和图 2-5 与图 2-6 所示的数据模型差不多。在得到这两款 GPS 接收机的实验数据后,下一步开始对整理好的数据进行分析。

对于第一款 GPS 接收机,我们观察图 2-4。从图 2-4 中我们可以看到,虽然这个 GPS 接收机的测量数据的系统误差和随机误差比较大,但是其数据的变化趋势比较缓慢。如果从图 2-4 上还是不太容易看出其特点,我们继续做进一步的处理。将图 2-4 所示的 GPS 的数据减去这些数据的平均值,减完后的结果如图 2-6 所示。

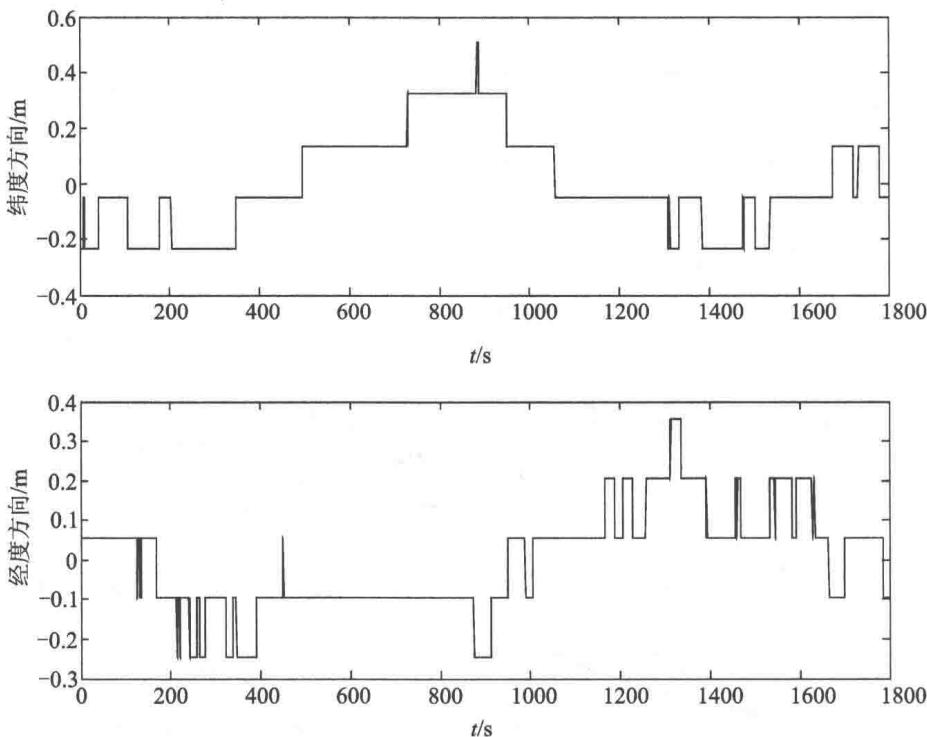


图 2-6 第一款 GPS 数据减去其平均值后的数据

从图 2-6 可以清楚地看到,不论这款 GPS 接收机的系统误差有多大,如果我们单纯地只考虑相对定位的话,其定位数据漂移得很慢。也就是说,对于第一款 GPS 接收机来说,其当前采样时刻的数据与下一采样时刻的数据相差不大。为了更加形象地表示这一特点,我们将该 GPS 接收机的定位数据的下一采样时刻的数据减去当前采样时刻的数据,其结果如图 2-7 所示。从图 2-7 中我们可以清楚地看到第一款 GPS 接收机的数据特点。对于第一款 GPS 接收机来说,虽然其绝对定位时的系统误差和随机误差相对来说比较大,无法单独提供导航信息,但是在用其后一采样时刻的数据减去当前采样时刻的数据时,我们发现其数据漂移得比较慢。也就是说,下一采样时刻的定位数据值与当前采样时刻的定位数据值变化不大。从图 2-7 中可以看出,在连续的几个采样时刻,其定位数据值保持稳定,而且当其变化时,其变化量也不大,一般保持在 1~2 个该 GPS 接收机的定位数据分辨率之间。在本次实验中,对于纬度方向来说为 0.1854 m,对于经度方向来说为

0.1503 m。图 2-7 中的数据选取的是比较有代表性的一段时间的数据。实际上,对于第一款 GPS 接收机,我们做了大量的实验,分别在不同性质的测量地点(包括水面上、空旷地带、周围有灌木丛、周围有高大树木和周围有高大建筑物等)、不同的时间点(包括凌晨、早晨、上午、中午、下午、晚间、半夜等)和不同的天气条件下(包括晴天、多云和阴天等)进行该 GPS 接收机定位数据采集。对采集到的测量结果均做了相同的处理,所得到的结果与图 2-7 类似。从大量的实验中我们得到了以上的结论。

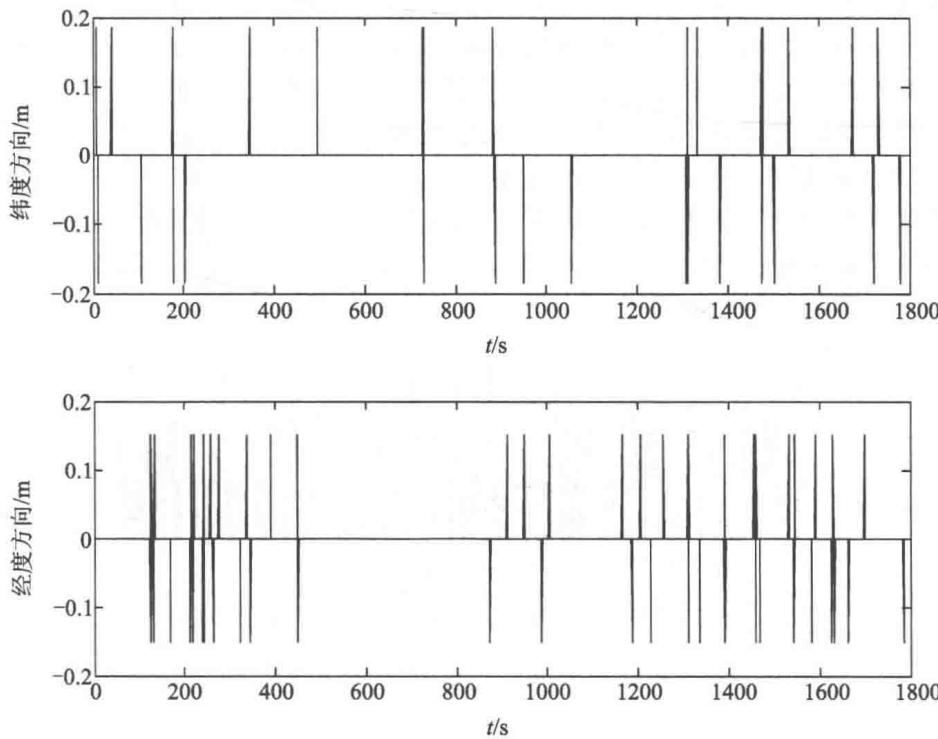


图 2-7 第一款 GPS 接收机的下一采样时刻数据减去当前采样时刻数据的结果

第二款 GPS 接收机是价格更为便宜的单频 GPS 接收机,经过大量实验,得到其绝对定位精度为 10~15 m,随机误差为 5~10 m。对于第二款 GPS 接收机,我们首先做了与第一款 GPS 接收机相同的工作,分别在不同性质的测量地点(包括水面上、空旷地带、周围有灌木丛、周围有高大树木和周围有高大建筑物等)、不同的时间点(包括凌晨、早晨、上午、中午、下午、晚间、半夜等)和不同的天气条件下(包括晴天、多云和阴天等)进行该 GPS 接收机定位数据的采集。对所得到的测量数据进行整理和分析后,得到了相应的结论。这里我们选取有代表性的一次测量数据。对于第二款 GPS 接收机,其测量的数据在减去其平均值后的数据如图 2-5 所示。从图 2-5 所示的数据中,我们无法一下子发现其数据的特点。图 2-5 中所

示的该 GSP 接收机的定位数据变化较大, 数据分布杂乱无章, 无法用第一款 GPS 接收机的分析方法进行分析。我们首先还是采用相对定位的思路, 用该 GPS 接收机的定位数据的下一采样时刻的定位数据减去当前采样时刻的定位数据, 其结果如图 2-8 所示。从图 2-8 中我们可以看出, 第二款 GPS 接收机的数据特点与第一款 GPS 接收机的数据特点不同, 第二款 GPS 接收机的数据误差范围更大, 相邻采样时刻的数据变化也更快些。但是, 从图 2-8 中我们依然可以看出一些特点。图 2-8 中所示的 GPS 接收机的定位数据变化虽然比较快, 但是相邻采样时刻的数据的变化并不太大。在进一步的分析中我们发现, 对于第二款 GPS 接收机来说, 其相邻采样时刻的数据的变化一般不超过该 GPS 接收机的定位分辨率的三倍。在纬度方向上其定位分辨率为 0.1854 m , 在精度方向上其定位分辨率为 0.1503 m 。继续观察图 2-8, 其展现了该 GPS 接收机定位数据随时间的变化趋势。我们观察到, 图 2-8 中的相邻定位数据是彼此存在着一些关系的。再观察图 2-6, 该数据的非线性非常严重。由于神经网络具有强大的模拟非线性函数的能力, 而图 2-8 中所示的数据间又有一定的相互关系, 我们考虑能否利用神经网络对第二款 GPS 接收机的定位数据进行模拟预测。

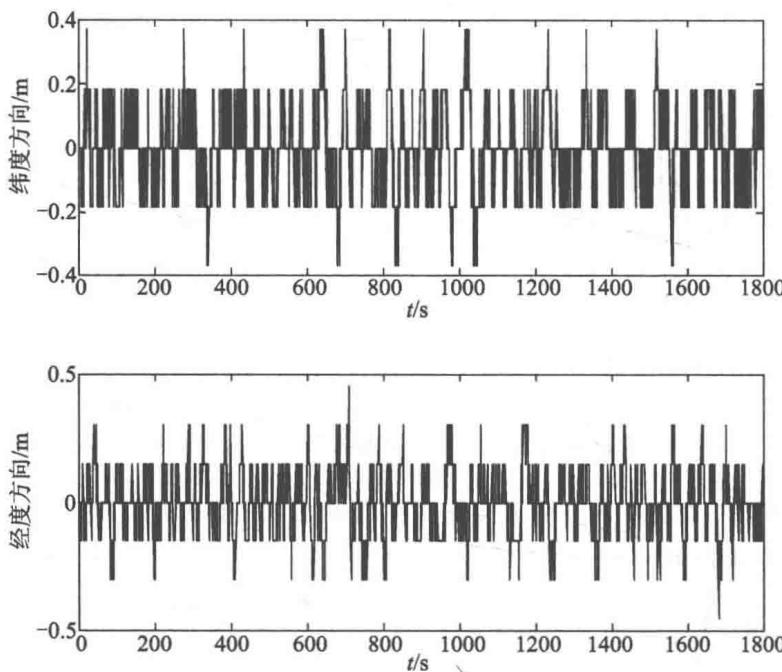


图 2-8 第二款 GPS 接收机的下一采样时刻数据减去当前采样时刻数据的结果

HDOP 是显示 GPS 接收机当前工作状况的重要参数, 它反映了 GPS 接收机所探测到的天空中 GPS 卫星的位置分布对 GPS 定位误差的影响。通过在同一地点的两次不同时间和天气情况下采集该民用单频 GPS 接收机的定位数据, 两次数据采集的时间均为 30 min, 每次采集共 1800 个测量值。表 2-1 展示的是这两次测

量中的 HDOP 值。表 2-2 展示了基于不同的 HDOP 值的每次测量值的标准偏差。从表 2-1 和 2-2 可以看出 HDOP 值可以在一定程度上反映了 GPS 接收机当前的定位精度。

表 2-1 两次测量中的 HDOP 值

第一次测量		第二次测量	
HDOP/m	采集数	HDOP/m	采集数
0~0.9	289	1.5	275
0.9~1.0	444	1.6	1210
1.0~1.1	667	1.7	215
>1.1	426	>1.7	89

表 2-2 两次测量的标准偏差 (m)

第一次测量			第二次测量		
HDOP	S-N	E-W	HDOP	S-N	E-W
0~0.9	0.9281	0.8858	1.5	1.1692	1.0181
0.9~1.0	1.3653	1.2006	1.6	2.1090	1.7128
1.0~1.1	1.3968	1.1744	1.7	1.8410	1.2318
>1.1	1.1644	1.0228	>1.7	2.0375	1.7307

3. 用神经网络进行 GPS 接收机定位数据模拟预测

人工神经网络 (artificial neural networks, 简写为 ANNs) 也简称为神经网络 (NNs) 或称为连接模型 (connectionist model), 它是一种模拟动物神经网络行为特征并进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度, 通过调整内部大量节点之间相互连接的关系, 从而达到处理信息的目的。神经网络是一门发展十分迅速的交叉学科, 它是由大量的处理单元组成非线性的大规模自适应动力系统。神经网络具有分布式存储、并行处理、高容错能力以及良好的自学习、自适应、联想等特点。常用到的神经网络的功能有以下几种: ① 联想记忆; ② 分类和聚类; ③ 优化计算。

人工神经网络是一个并行与分布式的信息处理网络结构, 该网络结构一般由许多个神经元组成, 每个神经元有一个单一的输出, 它可以连接到许多其他的神经元, 其输出有多个连接通路, 每个连接通路对应一个连接权系数。

严格来说, 神经网络是一个具有如下性质的有向图。

- ① 对于每个节点有一个状态变量 x_j 。
- ② 节点 i 到节点 j 有一个连接权系数 w_{ji} 。
- ③ 对于每个节点有一个阈值 θ_j 。

④对于每个节点定义一个变换函数 $f_i[x_i, w_{ji}, \theta_j (i \neq j)]$, 最常见的情形为 $f(\sum_i w_{ji} x_i - \theta_j)$ 。

到目前为止出现了很多种神经网络结构。图 2-9 展示了两个典型的神经网络结构, 左为前馈网络, 右为反馈网络。常见的结构有: 前馈神经网络、单层自反馈 Hopfield 神经网络、玻尔兹曼机神经网络、自组织神经网络、CMAC 神经网络、细胞神经网络等。其中 BP(back propagation) 神经网络具有良好的非线性系统模拟预测功能, 因此在本书中我们采用的是用 BP 神经网络去模拟预测第二款 GPS 接收机的定位数据。BP 神经网络模型是人工神经网络的重要模型之一。BP 神经网络是用误差传播学习算法(error back propagation, 即 BP 算法) 进行训练的多层次前馈神经网络。它有广泛的应用, 主要包括模式识别及分类、故障智能诊断、图像处理、函数拟合、最优预测等方面的应用。BP 神经网络的特点如下。

① 分布式的信息存储方式。神经网络是以各个处理器本身的状态和它们之间的连接形式存储信息的, 一个信息不是存储在一个地方, 而是按内容分布在整个网络上。网络上的某一处不是只存储一个外部信息, 而是存储了多个信息的部分内容。整个网络对多个信息加工后才存储到网络各处, 因此, 它是一种分布式存储方式。

② 大规模并行处理。BP 神经网络信息的存储与处理(计算)是合二为一的, 即信息的存储体现在神经元互连的分布上, 并以大规模并行分布方式处理为主, 比串行离散符号处理的现代数字计算机更优越。

③ 自学习性和自适应性。BP 神经网络各层直接的连接权值具有一定的可调性, 网络可以通过训练和学习来确定网络的权值, 呈现出很强的对环境的自适应能力和对外界事物的自学习能力。

④ 较强的鲁棒性和容错性。BP 神经网络分布式的信息存储方式, 使其具有较强的容错性和联想记忆功能, 这样如果某一部分的信息丢失或损坏, 网络仍能恢复出原来完整的信息, 系统仍能运行。

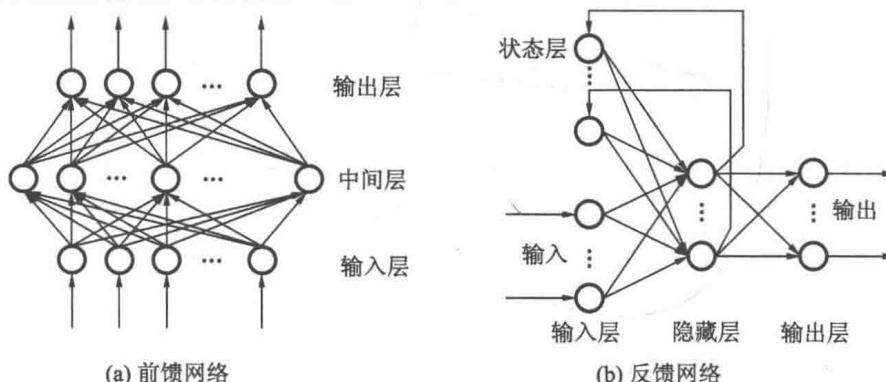


图 2-9 典型的神经元网络结构

人工神经网络具有广泛的应用领域。在所有的神经网络应用中, BP 神经网络所占比例在 80% 以上。BP 神经网络因其良好的非线性逼近能力和泛化能力以及使用的易适性而更是受到众多行业的青睐。BP 神经网络采用的反向传播算法 (BP 算法) 是目前在前馈神经网络中研究得最为成熟且应用最广的一种有导师学习算法。

BP 网络是一种单向传播的多层前馈网络, 它包含输入层、隐藏层和输出层, 如图 2-10 所示, BP 网络是目前应用较多的一种模型。BP 算法在层次型网络结构上采用误差逆传播学习方式, 学习过程由正向传播和误差逆传播组成。BP 算法的主要思想是把学习过程分为两个阶段: 第一阶段是正向传播过程, 输入信息从输入层经隐藏层逐层计算各单元的实际输出值, 每一层神经元的状态只对下一层神经元的状态产生影响; 第二阶段是反向传播过程, 若在输出层未能得到期望的输出值, 则逐层递归计算实际输出与期望输出之间的差值, 根据此误差修正前一层权值使误差信号趋向最小。它通过连续不断地在相对于误差函数斜率下降的方向上计算网络权值和偏差变化而逐渐逼近目标。每一次权值和误差的变化都与网络误差的影响成正比。

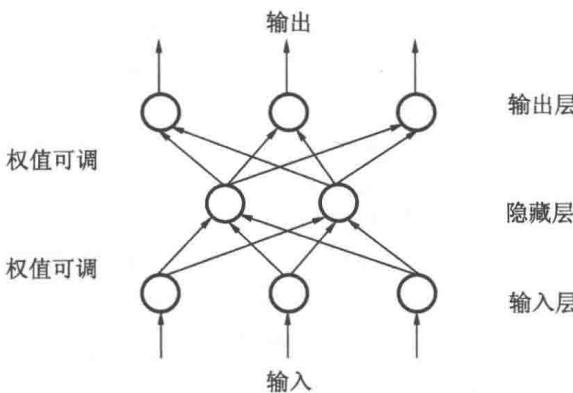


图 2-10 BP 神经网络示意图

BP 神经网络具有结构简单、可操作性强、能模拟任意的非线性输入/输出关系等优点, 目前已被广泛应用于模式识别、智能控制、预测、图像识别等领域。但是, BP 网络存在收敛速度慢、易陷入局部极小点这两个突出问题, 使其在应用方面受到了一定限制。其一, 收敛速度慢指的是, 由于其训练算法自身的局限性, 训练 BP 神经网络所需要的训练过程长, 不容易得到最佳结果。其二, 局部极小点指的是, 训练 BP 神经网络时, 最终结果收敛到一个不是最佳结果的局部极点上, 使得所得到的训练结果不好, 无法完成指定的任务。针对这两个问题, 很多针对 BP 神经网络训练方法的改进方法被提出来, 使得训练速度得到了一定的提高, 局部极小点的问题也得到了部分解决。关于 BP 神经网络训练算法的问题, 得到了广泛的关注, 但这不是我们这本书要讨论的主要内容。

神经网络模型的建立首先是选择神经网络的结构。在选定 BP 神经网络后,下一步就是要确定 BP 神经网络的具体结构。一般来说 BP 神经网络有三个主要部分:输入层、隐藏层和输出层。输入层和输出层都是一层结构,隐藏层可以是一层结构,也可以有多层结构。隐藏层的数目越多,神经网络的结构就越复杂,当然其能够模拟的非线性系统也就越复杂。然而,隐藏层的数目也不是越多越好。从训练计算的成本方面来考虑,隐藏层越多,训练的难度越大,训练所需的时间越久。根据以往的经验,对于大部分的非线性系统拟合要求,一个隐藏层基本上能够实现满意的拟合效果。选好隐藏层的数目后,下面应该选取每个层的节点数目。对于输入层来说,节点的数目等于训练所用数据的一组数据的个数,这取决于训练所用数据的选取。输出层的节点数目就是神经网络最终输出数据的个数。隐藏层的节点个数是我们要重点选取的。节点数目过多会导致神经网络的模型过于复杂,使得训练过程变得困难;节点数目偏少会使得神经网络的结构过于简单,无法实现模拟的任务。一般来说,隐藏层节点的选择都是先拟定一个适当的数目,然后在实验中进行相应的调整,以达到最佳的效果(既能够很好地完成拟合任务,又不会使训练计算的负担过大)。在神经网络的结构确定后,下一步是进行训练数据的准备。通常将数据分为两组:训练用数据和测试用数据。训练用数据是用来训练神经网络的,测试用数据是用来测试训练好的神经网络的模拟效果的。一般来说,要求数据能够覆盖所要模拟的系统的全部特点。训练用数据的数目和测试用数据的数目要求基本相差不多,这样能够测试所训练好的神经网络的广泛性。数据不要选取得太多,如果训练用数据选取得太多,会导致过度训练。过度训练的效果是使得训练好的神经网络的广泛性很差。也就是说,训练好的神经网络对于训练用数据可以达到很好的模拟效果,而对于非训练用数据(如测试用数据)则模拟效果很差。训练用数据也不能太少,如果训练用数据太少,其数据无法覆盖所要模拟模型的全部特性,则无法达到训练的效果。所以,训练和测试用的数据要适当的选择。总的来说,神经网络的结构和训练数据的选择是一个实验的过程,要根据实际的实验情况做适当的调整,直至达到最佳的模拟效果。先根据经验来选择一个神经网络的结构和训练用数据,然后进行神经网络的训练和测试。如果达不到训练的要求,就要重新选取训练和测试用的数据,重新确定隐藏层节点的数目,直至重新选取神经网络的结构,达到满意的模拟效果为止。

由于该民用单频 GPS 接收机的相邻输出信息之间存在一定的关系,以及 HDOP 值的重要性,考虑采用神经网络基于前若干个时刻的 GPS 接收机的输出和相应的 HDOP 值进行当前时刻 GPS 数据的预估。这里我们选用的是 BP 神经网络。图 2-11 展示的是在本书中我们所选用的神经网络的输入、输出的结构关系。该神经网络拥有三个层:一个输入层、一个隐藏层和一个输出层(M_p 为输入层到隐藏层的权值, M_q 为隐藏层到输出层的权值)。图 2-12 展示的是该神经网络模型

的结构。 $x(k-1)$ 、 $x(k-2)$ 和 $x(k-3)$ 是前三个采样时刻的 GPS 接收机的输出(经度值或纬度值), $HDOP(k-1)$ 、 $HDOP(k-2)$ 和 $HDOP(k-3)$ 是前三个采样时刻的 HDOP 值, $\hat{x}(k)$ 是当前时刻 GPS 接收机的输出的预估值(经度值或纬度值)。

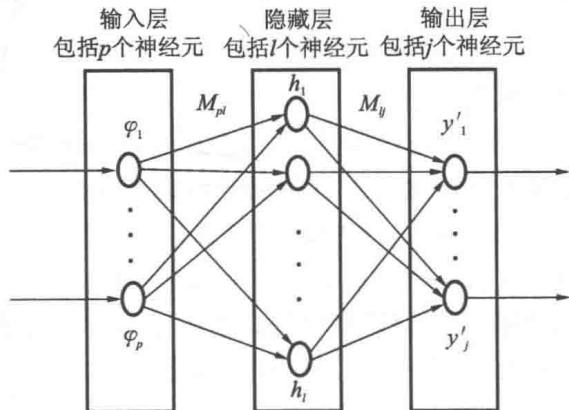


图 2-11 所选神经网络输入、输出的关系简图

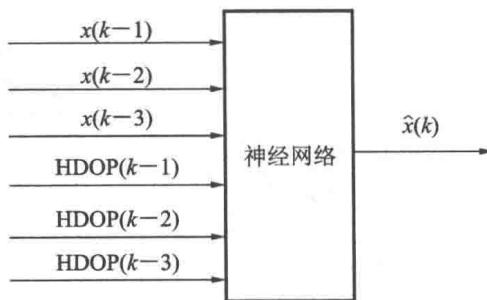


图 2-12 所用神经网络的结构简图

从图 2-12 中我们可以看到所用的神经网络的输入数据包括 6 项。对于本书中的内容来说,实际上是 9 项数据,包括:前三个采样时刻的纬度数据、前三个采样时刻的经度数据和前三个采样时刻的 HDOP 值。如前所述,HDOP 值决定于 GPS 接收机所探测到的 GPS 导航卫星的相对位置对定位精度的影响程度。该神经网络输出的是当前采样时刻的纬度数据和经度数据。也就是说,通过该神经网络,我们要用前三个采样时刻的 GPS 接收机输出的纬度、精度和 HDOP 值等数据来预测当前采样时刻的 GPS 接收机输出的纬度和精度的数据。

经过不断的选取和测试,我们最终得到了取得满意模拟预测效果的神经网络的结构。所选择的神经网络有三个层次:一个输入层,包括 9 个节点;一个输出层,包括 2 个节点;一个隐藏层,包括 5 个节点。隐藏层的传递函数选取为“TANSIG”函数($f_h(x) = \tanh(x)$),输出层的传递函数选取为“PURELIN”函数($f_o(x) = x$)。

神经网络的输入信号是前三个采样时刻 GPS 的输出(经度值和纬度值)与相应的 HDOP 值,神经网络的输出是当前采样时刻的 GPS 接收机的输出(经度值和纬度值)。用于调整神经网络各项权值的误差信号是当前时刻真实 GPS 接收机的输出与神经网络输出的差值。实际中多次在不同地点、时间和天气情况下的 GPS 的采集结果被用于进行该神经网络的测试,均得到了满意的预估结果。这里仅就一次采集所得到的数据进行训练和测试。在一个固定点上首先采集 1860 个采样周期的 GPS 数据。其中前 998 个采样数据用来进行神经网络的训练,而剩下的 862 个采样数据用来对训练好的神经网络进行测试。该训练和测试的 GPS 接收机的预测数据结果展示在图 2-13 和图 2-14 中。从图 2-13 和图 2-14 中,我们可以看到,我们所得到的神经网络可以很好地基于前三个采样时刻的纬度数据、经度数据和 HDOP 值等数据来预测第二款 GPS 接收机的当前时刻的纬度数据和经度数据(图 2-13、图 2-14 中 GPS 数据与神经网络输出近似重合)。这样,我们得到了两款 GPS 接收机的数据特点。在掌握了其数据特点后,下一步的工作就是要设计一种信息融合算法,使其能够很好地融合来自航迹推算导航系统的导航信息和所选民用单频 GPS 接收机的导航信息。

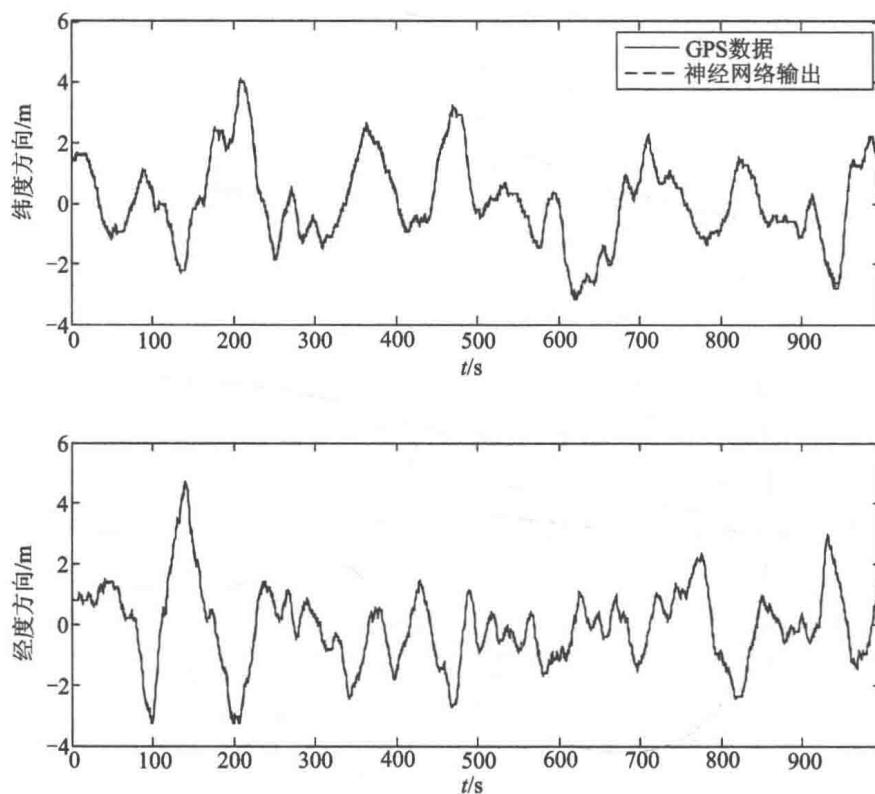


图 2-13 训练用数据的神经网络预测结果

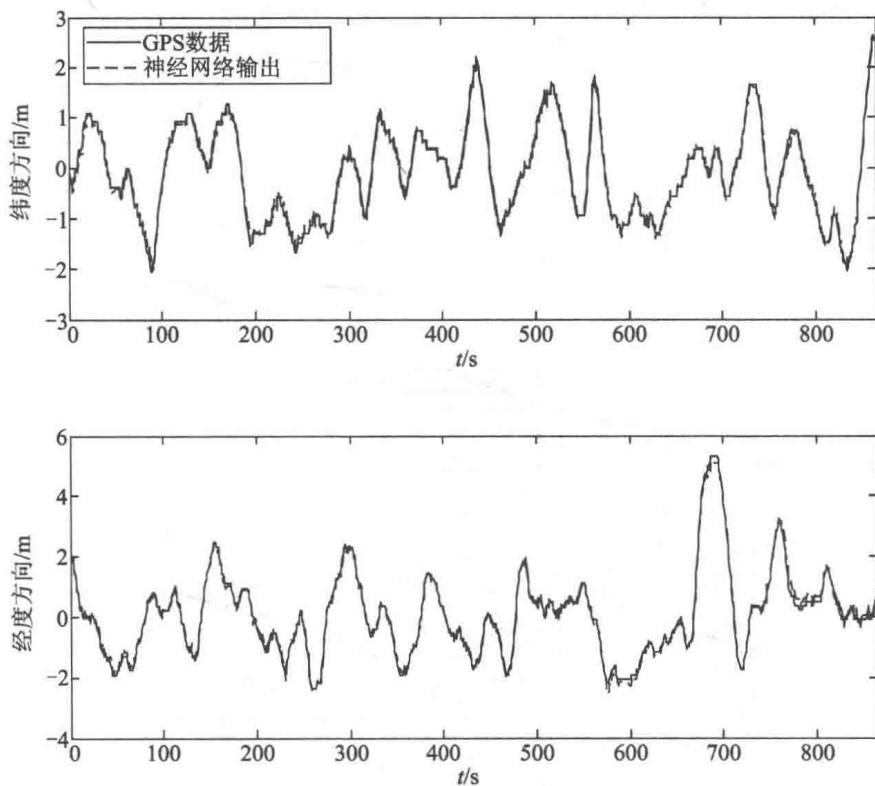


图 2-14 测试用数据的神经网络预测结果

2.3 导航信息融合算法的设计

信息融合技术是近年来十分热门的研究课题,它结合了控制理论、信号处理、人工智能、概率和统计等的发展。信息融合是指综合来自多个传感器的感知数据,以产生更可靠、更准确或更精确的信息的方法。经过融合的多传感器系统能完善地、精确地反映检测对象特性,消除信息的不确定性,提高传感器的可靠性。经过融合的多传感器信息具有以下特性:提高信息的冗余性、信息的互补性、信息的实时性和信息的低成本性。信息融合的方法有很多种,针对不同特点的系统需要应用不同的信息融合方法。图 2-15 展示了一些常用的信息融合方法。

信息融合的方法是多传感器信息融合的最重要的部分,由于其应用上的复杂性和多样性,决定了信息融合的研究内容极其丰富,涉及的基础理论较多。多传感器信息融合方法可以分为以下四类:估计方法、分类方法、推理方法和人工智能方法。其中分类方法主要有参数模板法和聚类分析。无监督或自组织学习算法诸如学习向量量化法(learning vector quantization, LVQ)、K-均值聚类(K-means clus-

tering) 算法、Kohonen 特性图 (Kohonen feature map) 等也常用作多传感器数据的分类。

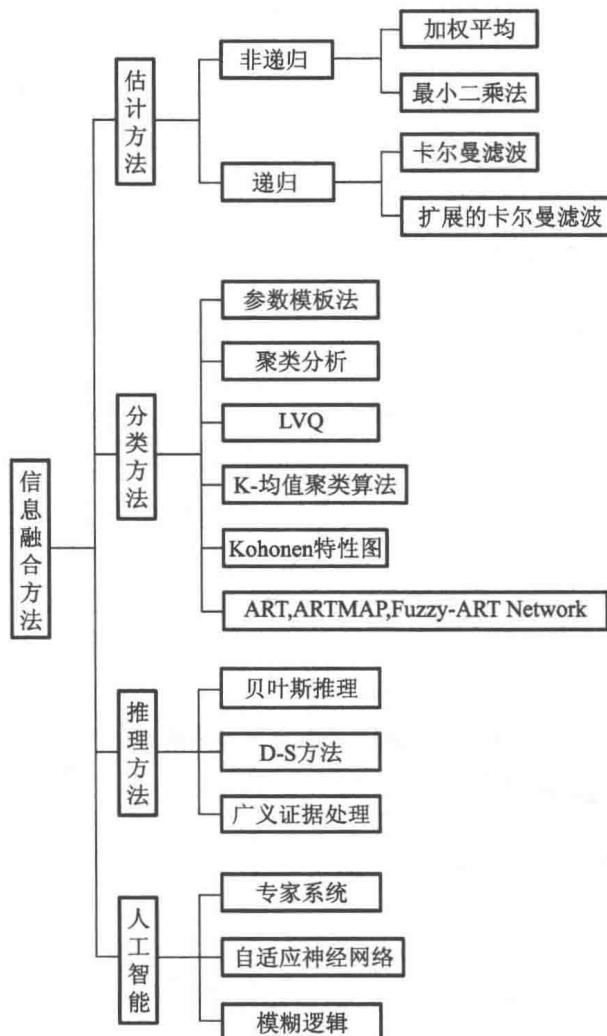


图 2-15 一些常用的信息融合方法

K-均值聚类算法是最常用的无监督学习算法之一,而自适应 K-均值方法的更新规则成为了 Kohonen 特性图的基础。此外自适应共振理论(ART)、自适应共振理论映射(ARTMAP) 和模糊自适应共振理论网络(fuzzy-ART network) 等以自适应的方法进行传感器融合。它们能够自动调整权值并且能在环境变化和输入漂移的情况下保持稳定。

人工智能方法可分为专家系统、自适应神经网络和模糊逻辑,对融合大量的传感器信息,在非线性和不确定的场合颇有优势。专家系统是一种基于人工智能的计算机信息系统。自适应神经网络是一个具有高度非线性的超大规模连续时间自

适应信息处理系统。在多传感器系统中,各信息源提供的环境信息都具有一定程度的不确定性,对这些不确定信息的融合过程实质上是一个不确定性推理过程。神经网络可根据当前系统所接收的样本的相似性,确定分类标准。同时可以采用特定的学习算法来获取知识,得到不确定性推理机制。模糊逻辑是多值逻辑,它允许将传感器信息融合过程中的不确定性直接表示在推理过程中。由于模糊集表达了一个不确定概念,应用模糊理论并结合其他手段(如神经网络),可以取得更好的融合结果。

针对本书中航迹推算导航系统和所选的两种民用单频 GPS 接收机系统的特点,经过仔细分析,我们决定采用基于贝叶斯理论和卡尔曼滤波的思想来设计导航信息融合算法。

贝叶斯定理(Bayes' theorem),是概率论中的一个结果,它跟随机变量的条件概率以及边缘概率分布有关。在有些关于概率的解说中,贝叶斯定理(贝叶斯更新)能够告知我们如何利用新证据修改已有的看法。

通常,事件 A 在事件 B(发生)的条件下的概率,与事件 B 在事件 A(发生)的条件下的概率是不一样的。然而,这两者是有确定的关系,贝叶斯定理就是对这种关系的陈述。

贝叶斯定理是关于随机事件 A 和 B 的条件概率和边缘概率的一则定理。

如果两个事件 A 和 B 不是互相独立的,并且知道事件 B 中的一个事件已经发生,我们就能得到关于 $P(A)$ 的信息。这反映为 A 在 B 中的条件概率,记为 $P(A | B)$:

$$P(A | B) = \frac{P(AB)}{P(B)} \quad (2-4)$$

无条件概率 $P(A)$ 通常称为先验概率,而条件概率 $P(A | B)$ 通常称为后验概率。

注意:条件可以在任何一个中发生:

$$P(AB) = P(A | B)P(B) = P(B | A)P(A) \quad (2-5)$$

条件概率是概率的有效测度,所以它满足所有的基本公理。

如果 A 和 B 是独立的,则

$$P(A | B) = P(A) \quad (2-6)$$

以上所列公式中, $P(A | B)$ 是在 B 发生的情况下 A 发生的可能性。在贝叶斯定理中,每个名词都有约定俗成的名称:

(1) $P(A)$ 是 A 的先验概率或边缘概率,之所以称为“先验”是因为它不考虑任何 B 方面的因素。

(2) $P(A | B)$ 是已知 B 发生后 A 的条件概率,也由于得自 B 的取值而被称作 A 的后验概率。

(3) $P(B|A)$ 是已知 A 发生后 B 的条件概率, 也由于得自 A 的取值而被称作 B 的后验概率。

(4) $P(B)$ 是 B 的先验概率或边缘概率, 也作标准化常量 (normalized constant)。

按这些术语, 贝叶斯定理可表述为

$$\text{后验概率} = (\text{相似度} \times \text{先验概率}) / \text{标准化常量}$$

也就是说, 后验概率与先验概率和相似度的乘积成正比。

另外, 比例 $P(B|A)/P(B)$ 也有时被称为标准相似度 (standardised likelihood), 贝叶斯定理可表述为

$$\text{后验概率} = \text{标准相似度} \times \text{先验概率}$$

1960 年, 卡尔曼发表了著名的用递归方法解决离散数据线性滤波问题的论文。从那以后, 得益于数字计算技术的进步, 卡尔曼滤波器已成为推广研究和应用的主题, 尤其是在自主或协助导航领域。卡尔曼滤波器由一系列递归数学公式描述。它们提供了一种高效可计算的方法来估计过程的状态, 并使估计均方误差最小。卡尔曼滤波器应用广泛且功能强大, 它可以估计信号的过去和当前状态, 甚至能估计将来的状态, 即使并不知道模型的确切性质。

卡尔曼滤波器用于估计离散时间过程的状态变量 $x \in \mathbf{R}^n$ 。这个离散时间过程由以下离散随机差分方程描述:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (2-7)$$

定义观测变量 $z \in \mathbf{R}^m$, 得到量测方程:

$$z_k = \mathbf{H}x_k + v_k \quad (2-8)$$

随机信号 w_k 和 v_k 分别表示过程激励噪声和观测噪声。假设它们为相互独立、正态分布的白色噪声:

$$p(w) \sim N(0, Q) \quad (2-9)$$

$$p(v) \sim N(0, R) \quad (2-10)$$

实际系统中, 过程激励噪声协方差矩阵 Q 和观测噪声协方差矩阵 R 可能会随每次迭代计算而变化。但在这儿我们假设它们是常数。

当控制函数 u_{k-1} 或过程激励噪声 w_{k-1} 为零时, 差分方程(2-7)中的 n 阶增益矩阵 A 将上一时刻 $k-1$ 的状态线性映射到当前时刻 k 的状态。实际中 A 可能随时间变化, 但在这儿假设为常数。 $n \times l$ 矩阵 B 代表可选的控制输入 $u \in \mathbf{R}^l$ 的增益。量测方程(2-8)中的 $m \times n$ 矩阵 H 表示状态变量 x_k 对测量变量 z_k 的增益。实际中 H 可能随时间变化, 但在这儿假设为常数。

定义 $\hat{x}_k^- \in \mathbf{R}^n$ ($-$ 代表先验, $\hat{\cdot}$ 代表估计) 为在已知第 k 步以前状态情况下第 k 步的先验状态估计。定义 $\hat{x}_k \in \mathbf{R}^n$ 为已知测量变量 z_k 时第 k 步的后验状态估计。由此定义先验估计误差和后验估计误差:

$$\begin{aligned} e_k^- &= x_k - \hat{x}_k^- \\ e_k &= x_k - \hat{x}_k \end{aligned} \quad (2-11)$$

先验估计误差的协方差为

$$P_k^- = E[e_k^- e_k^{-T}] \quad (2-12)$$

后验估计误差的协方差为

$$P_k = E[e_k e_k^T] \quad (2-13)$$

式(2-14)构造了卡尔曼滤波器的表达式:先验估计 \hat{x}_k^- 和加权的测量变量 z_k 及其预测 $H\hat{x}_k^-$ 之差的线性组合构成了后验状态估计 \hat{x}_k 。

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}(z_k - H\hat{x}_k^-) \quad (2-14)$$

式(2-14)中测量变量及其预测之差($z_k - H\hat{x}_k^-$)被称为测量过程的革新或残余。残余反映了预测值和实际值之间的不一致程度。残余为零表明二者完全吻合。

式(2-14)中 $n \times m$ 矩阵 \mathbf{K} 称为残余的增益或混合因数,作用是使式(2-13)中的后验估计误差协方差最小。可以通过以下步骤计算 \mathbf{K} :首先将式(2-14)代入 e_k 的定义式,再将 e_k 代入式(2-13)中,求得期望后,将式(2-13)中的 P_k 对 \mathbf{K} 求导,并使一阶导数为零从而解得 \mathbf{K} 值。 \mathbf{K} 的一种表示形式为

$$\mathbf{K} = P_k^- \mathbf{H}^T (\mathbf{H} P_k^- \mathbf{H}^T + \mathbf{R})^{-1} = \frac{P_k^- \mathbf{H}^T}{\mathbf{H} P_k^- \mathbf{H}^T + \mathbf{R}} \quad (2-15)$$

由式(2-15)可知,观测噪声协方差 R 越小,残余的增益 K 越大。特别的, R 趋向于零时,有

$$\lim_{R_k \rightarrow 0} K_k = \mathbf{H}^{-1} \quad (2-16)$$

另一方面,先验估计误差协方差 P_k^- 越小,残余的增益 K 越小。特别地, P_k^- 趋向于零时,有

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (2-17)$$

增益 K 的另一种解释是随着测量噪声协方差 R 趋于零,测量变量 z_k 的权重越来越大,而 z_k 的预测 $H\hat{x}_k^-$ 的权重越来越小。另一方面,随着先验估计误差协方差 P_k^- 趋于零,测量变量 z_k 的权重越来越小,而 z_k 的预测 $H\hat{x}_k^-$ 的权重越来越大。

卡尔曼滤波器用反馈控制的方法估计过程状态。滤波器估计过程某一时刻的状态,然后以(含噪声的)测量变量的方式获得反馈。因此,卡尔曼滤波器可分为两个部分:时间更新方程和测量更新方程。时间更新方程负责及时向前推算当前状态变量和误差协方差估计的值,以便为下一个时间状态构造先验估计。测量更新方程负责反馈。也就是说,它将先验估计和新的测量变量结合以构造改进的后验估计。

时间更新方程也可视为预估方程,测量更新方程可视为校正方程。最后的估计算法可视为一种具有数值解的预估-校正算法,如图 2-16 所示。

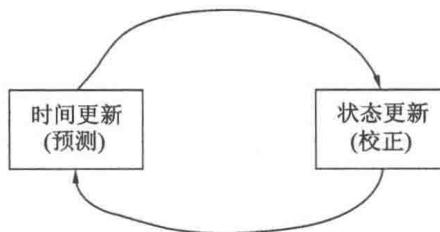


图 2-16 离散卡尔曼滤波器循环更新图

表 2-3 和表 2-4 分别给出了时间更新方程和测量更新方程的具体形式。

表 2-3 离散卡尔曼滤波器时间更新方程

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (2-18)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2-19)$$

表 2-4 离散卡尔曼滤波器测量更新方程

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (2-20)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad (2-21)$$

$$P_k = (I - K_k H)P_k^- \quad (2-22)$$

测量更新方程,首先做的是计算卡尔曼增益 K_k ;其次便测量输出以获得 z_k ,然后按式(2-21)产生状态的后验估计;最后按式(2-22)估计状态的后验协方差。

计算完时间更新方程和测量更新方程后,将整个过程再次重复。上一次计算得到的后验估计被作为下一次计算的先验估计。这种递归推算是卡尔曼滤波器最吸引人的特性之一。图 2-17 将表 2-3 和表 2-4 结合在一起,显示了滤波器的整个操作流程。

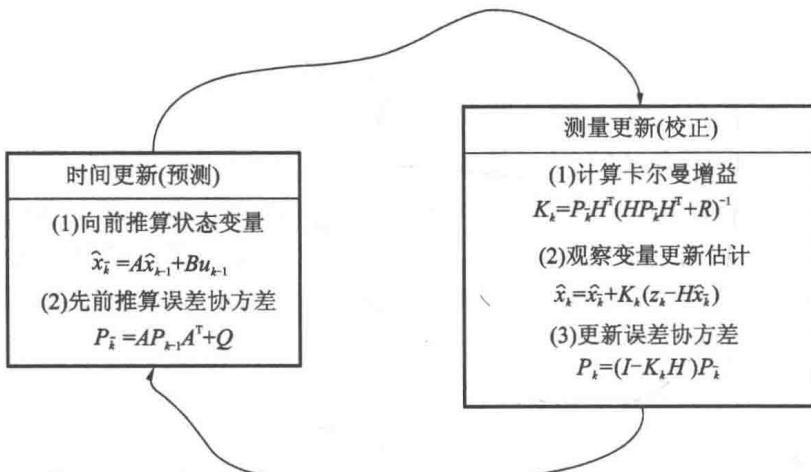


图 2-17 卡尔曼滤波器工作原理图

在仔细研究了贝叶斯理论和卡尔曼滤波后,再结合我们前面得到的两款 GPS 接收机的数据特点,下一步开始针对每一款 GPS 接收机设计航迹推算导航系统和 GPS 系统的导航信息融合算法。

1) 基于第一款 GPS 接收机数据特点的 GPS/DR 数据融合算法的设计

采用图 2-2 所示的第一款民用单频 GPS 接收机和一个简单的 DR 系统来为室外移动机器人提供导航信息。DR 系统是一种非常常用的机器人导航系统。对于一个在二维空间中运行的移动机器人来说,在出发点已知的情况下,利用 DR 系统可以计算在任意时刻的机器人的位置。在本书中电子罗盘和两个轴编码器用来给出该移动机器人的方向和速度。图 2-18 所示为一个轮式移动机器人的简图。本书中考虑的移动机器人有两个驱动轮(半径为 r)和一个辅助轮。图 2-18 中,点 $H(x_H, y_H)$ 是两个驱动轮之间轴的中点,设为参考坐标系 $\{X_H, Y_H\}$ 的坐标原点。点 $C(x_C, y_C)$ 是移动机器人的重心, c 为点 H 和点 C 之间的距离, l 是机器人后轮轴的长度。机器人的方向角 θ 和线速度 v 可以用公式(2-23)来解算移动机器人的位置坐标。

$$\begin{cases} \dot{x}_H = v(t) \cos(\theta) \\ \dot{y}_H = v(t) \sin(\theta) \end{cases} \quad (2-23)$$

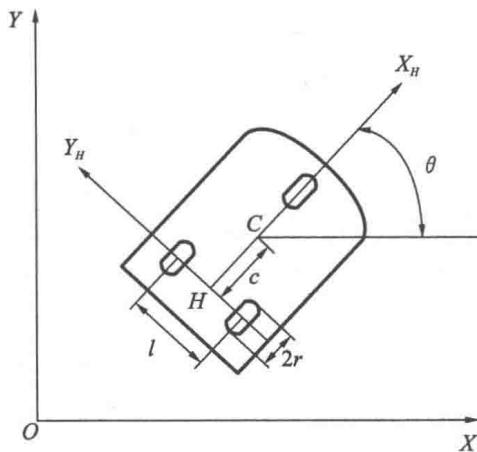


图 2-18 轮式移动机器人结构简图

在得出所选的第一款民用单频 GPS 接收机导航数据特点的基础上,下一步要进行 GPS/DR 综合导航系统导航信息融合算法的设计。DR 系统可以提供短期准确的导航信息,但是其误差会随着时间的增加而不断累积。另一方面,民用单频 GPS 接收机的绝对误差很大,但是书中所选用的第一款民用单频 GPS 接收机的数据有明显的可以利用的特点。对于第一款民用单频 GPS 接收机来说,其数据特点为:虽然其绝对定位时的系统误差和随机误差相对来说比较大,无法单独地提供导

航信息,但是在用其后一采样时刻的数据减去当前采样时刻的数据时,我们发现其数据漂移得比较慢。也就是说下一采样时刻的定位数据值与当前采样时刻的定位数据值变化不大。从图 2-7 中可以看出,在连续的几个采样时刻,其定位数据值保持稳定,而当变化时,其变化量也不大,一般保持在 1~2 个该 GPS 接收机的定位数据分辨率。在本次实验中,对于纬度方向来说为 0.1854 m,对于经度方向来说为 0.1503 m。我们假设方向 x 代表纬度方向、方向 y 代表经度方向。假设 DR 系统提供的当前移动机器人的位置坐标的可信度在 x 方向为 Q_x 和在 y 方向为 Q_y 。假设基于所选第一种民用单频 GPS 接收机所提供的移动机器人位置坐标的可信度在 x 方向为 R_x 和在 y 方向为 R_y 。由于 DR 系统短期的导航精度非常高,而民用单频 GPS 接收机的导航信息的误差要大得多,因此 Q_x 和 Q_y 的值要远远大于 R_x 和 R_y 的值。基于如上所述的所选第一款民用单频 GPS 接收机和 DR 系统的数据特点,进行 GPS/DR 综合导航系统导航数据融合算法的设计。所设计的导航数据融合算法可以用图 2-19 来简要描述。

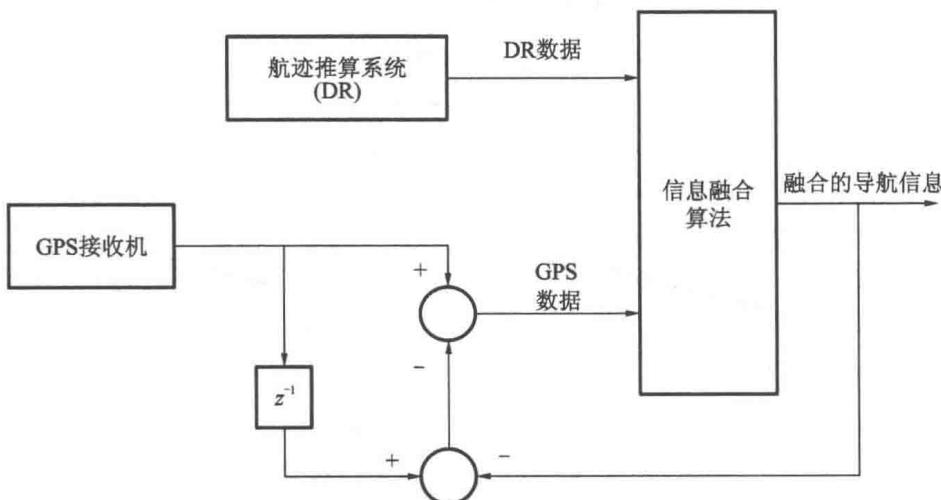


图 2-19 基于所选第一款民用单频 GPS 接收机所设计的 GPS/DR 导航信息融合算法简图

由于所设计的 GPS/DR 综合导航系统是基于相对定位进行设计的,所以需假设在移动机器人的出发点固定一个假想的 GPS 接收机,该 GPS 接收机与所选的第一款民用单频 GPS 接收机的数据特点相同。该综合导航系统导航信息融合算法如下所述。

①在出发点上,在机器人没有出发时先采集一次 GPS 接收机的数据 $GPS_x(0)$ 和 $GPS_y(0)$ 。其中 $GPS_x(0)$ 和 $GPS_y(0)$ 是 GPS 接收机的输出,也就是在时刻 $t=0$ s 时 GPS 接收机给出的出发点的经度信息和纬度信息。假设在时刻 $t=0$ s 时,假想的 GPS 接收机与实际固定在移动机器人上的真实的 GPS 接收机有着同

样的输出。也就是说 $GPS_{L_x}(0) = GPS_x(0)$ 和 $GPS_{L_y}(0) = GPS_y(0)$ 。其中, $GPS_{L_x}(0)$ 和 $GPS_{L_y}(0)$ 是假想的 GPS 接收机在时刻 $t=0$ s 时的输出数据, 也就是经度信息和纬度信息。

②在出发点上移动机器人开始出发。由于所选的第一款民用单频 GPS 接收机每一秒钟输出一次数据, 因此 GPS/DR 综合导航系统的导航信息融合的时间间隔为 1 s。在时刻 $t=1$ s 时, 通过 DR 系统可以得到移动机器人的坐标 $x_{DR}(1)$ 和 $y_{DR}(1)$ 。其中, $x_{DR}(1)$ 和 $y_{DR}(1)$ 是在时刻 $t=1$ s 时基于 DR 系统的移动机器人的坐标值。假设坐标值 $x_{DR}(1)$ 和 $y_{DR}(1)$ 的可信度分别为 Q_x 和 Q_y 。

③在时刻 $t=1$ s 时基于 GPS 接收机的数据可以得到 GPS 接收机的输出数据, 也就是移动机器人的经度值和纬度值, 即 $GPS_x(1)$ 和 $GPS_y(1)$ 。

④现在考虑固定在出发点上的假想 GPS 接收机在时刻 $t=1$ s 时的输出。如前所述, 由于所选的第一款民用单频 GPS 接收机的输出变化很慢, 相邻两个采样时刻的输出值几乎不变, 我们可以基于时刻 $t=0$ s 时的假想 GPS 接收机的输出值 $GPS_{L_x}(0)$ 和 $GPS_{L_y}(0)$ 来预测时刻 $t=1$ s 时假想 GPS 接收机的输出值: $GPS_{L_x}(1) = \hat{GPS}_{L_x}(0)$ 和 $GPS_{L_y}(1) = \hat{GPS}_{L_y}(0)$ 。(其中符号“ $\hat{\cdot}$ ”表示该值为假想 GPS 接收机输出值的预测值。)

⑤基于上述的数据, 可以得到在时刻 $t=1$ s 时基于 GPS 接收机所得到的移动机器人的坐标值。该坐标值可以用当前时刻固定在移动机器人上的真实 GPS 接收机的输出值和固定在出发点上的假想 GPS 接收机的输出值的差来表示: $x_{GPS}(1) = GPS_x(1) - \hat{GPS}_{L_x}(1)$ 和 $y_{GPS}(1) = GPS_y(1) - \hat{GPS}_{L_y}(1)$ 。其中, $x_{GPS}(1)$ 和 $y_{GPS}(1)$ 分别是时刻 $t=1$ s 时基于 GPS 接收机的移动机器人的坐标值。

⑥设置基于 GPS 接收机的移动机器人坐标值 $x_{GPS}(1)$ 和 $y_{GPS}(1)$ 的可信度 R_x 和 R_y 。如前所述, 由于 DR 系统的短期误差非常小, 因此可以利用 $x_{GPS}(1)$ 和 $x_{DR}(1)$ 的差值、 $y_{GPS}(1)$ 和 $y_{DR}(1)$ 的差值来设置 $x_{GPS}(1)$ 和 $y_{GPS}(1)$ 的可信度。具体来说, 对于纬度方向: 如果 $|x_{GPS}(1) - x_{DR}(1)| \leq 0.1854$ m, 则 $x_{GPS}(1)$ 的可信度设置为 R_{x1} ; 如果 $|x_{GPS}(1) - x_{DR}(1)| \leq 0.1854 \times 2$ m, 则 $x_{GPS}(1)$ 的可信度设置为 R_{x2} ; 若 $|x_{GPS}(1) - x_{DR}(1)| \geq 0.1854 \times 2$ m, 这说明由于发生了一些突发情况使得 GPS 的数据跳动比较大, 这时 GPS 工作在一种不良的工作状况下, 从而导致 GPS 接收机的导航精度下降很多, 在这种情况下 $x_{GPS}(1)$ 的可信度设置为 R_{x3} 。对于经度方向: 如果 $|y_{GPS}(1) - y_{DR}(1)| \leq 0.1503$ m, 则 $y_{GPS}(1)$ 的可信度设置为 R_{y1} ; 如果 $|y_{GPS}(1) - y_{DR}(1)| \leq 0.1503 \times 2$ m, 则 $y_{GPS}(1)$ 的可信度设置为 R_{y2} ; 若 $|y_{GPS}(1) - y_{DR}(1)| \geq 0.1503 \times 2$ m, 这说明由于发生了一些突发情况使得 GPS 的数据跳动比较大, 这时 GPS 工作在一种不良的工作状况下, 从而导致 GPS 接收机的导航精度下降很多, 在这种情况下 $y_{GPS}(1)$ 的可信度设置为 R_{y3} 。在此就 $x_{GPS}(1)$ 和 $y_{GPS}(1)$ 的可信度做如下的说明: 所设计的 GPS/DR 综合导航系统导航信息融

合的时间间隔为 1 s, 而由于在 1 s 的时间内 DR 系统的导航精度要远远大于民用单频 GPS 接收机的导航精度, 因此 $x_{\text{DR}}(1)$ 与 $y_{\text{DR}}(1)$ 的可信度 Q_x 与 Q_y 要远远大于 $x_{\text{GPS}}(1)$ 与 $y_{\text{GPS}}(1)$ 的可信度 R_{xi} 与 R_{yi} ($i=1, 2, 3$)。并且它们的关系可以用公式(2-24)来表示。

$$\begin{aligned} R_{x1} &> R_{x2} \gg R_{x3} \\ R_{y1} &> R_{y2} \gg R_{y3} \end{aligned} \quad (2-24)$$

⑦在时刻 $t=1$ s 时, 民用单频 GPS 接收机和 DR 系统的导航信息融合结果的解算, 可以用公式(2-25)和公式(2-26)进行计算。

$$x(1) = \frac{Q_x}{Q_x + R_x} x_{\text{DR}}(1) + \frac{R_x}{Q_x + R_x} x_{\text{GPS}}(1) \quad (2-25)$$

$$y(1) = \frac{Q_y}{Q_y + R_y} y_{\text{DR}}(1) + \frac{R_y}{Q_y + R_y} y_{\text{GPS}}(1) \quad (2-26)$$

其中: $x(1)$ 和 $y(1)$ 是 GPS 和 DR 综合导航系统导航信息融合后的移动机器人的位置坐标。移动机器人的方向信息可以由电子罗盘来提供。

⑧当前时刻固定在出发点的假想 GPS 接收机的输出值可以由公式(2-27)给出。

$$\begin{aligned} \text{GPS}_{I_x}(1) &= \text{GPS}_x(1) - x(1) \\ \text{GPS}_{I_y}(1) &= \text{GPS}_y(1) - y(1) \end{aligned} \quad (2-27)$$

⑨在时刻 $t=k$ s ($k=2, 3, \dots$) 时, 通过 DR 系统可以得到基于 DR 系统的移动机器人的坐标值: $x_{\text{DR}}(k)$ 和 $y_{\text{DR}}(k)$ 。设置 $x_{\text{DR}}(k)$ 和 $y_{\text{DR}}(k)$ 的可信度分别为 Q_x 和 Q_y 。

⑩从真实的 GPS 接收机可以得到当前时刻的 GPS 的输出值 $\text{GPS}_x(k)$ 和 $\text{GPS}_y(k)$ 。

⑪类似于步骤①, 可以进行解算当前时刻的固定在出发点上的假想 GPS 接收机的输出值 $\text{GPS}_{I_x}(k) = \text{GPS}_{I_x}(k-1)$ 和 $\text{GPS}_{I_y}(k) = \text{GPS}_{I_y}(k-1)$ 。

⑫类似于步骤⑤, 可以计算基于 GPS 的移动机器人的坐标值: $x_{\text{GPS}}(k) = \text{GPS}_x(k) - \text{GPS}_{I_x}(k)$ 和 $y_{\text{GPS}}(k) = \text{GPS}_y(k) - \text{GPS}_{I_y}(k)$ 。

⑬类似于步骤⑥, 进行 $x_{\text{GPS}}(k)$ 和 $y_{\text{GPS}}(k)$ 的可信度的设计。

⑭计算在时刻 $t=k$ s 时 GPS 和 DR 综合导航系统导航信息融合的结果, 也就是移动机器人的当前时刻坐标值, 可以用公式(2-28)和公式(2-29)来解算。

$$x(k) = \frac{Q_x}{Q_x + R_x} x_{\text{DR}}(k) + \frac{R_x}{Q_x + R_x} x_{\text{GPS}}(k) \quad (2-28)$$

$$y(k) = \frac{Q_y}{Q_y + R_y} y_{\text{DR}}(k) + \frac{R_y}{Q_y + R_y} y_{\text{GPS}}(k) \quad (2-29)$$

⑮当前时刻固定在出发点上的假想 GPS 接收机的输出值为

$$\begin{aligned} \text{GPS}_{I_x}(k) &= \text{GPS}_x(k) - x(k) \\ \text{GPS}_{I_y}(k) &= \text{GPS}_y(k) - y(k) \end{aligned} \quad (2-30)$$

如上所述就是基于所选第一款民用单频 GPS 接收机和 DR 系统的导航数据特点所设计的 GPS/DR 综合导航系统导航信息融合算法。

2) 第一种 GPS/DR 综合导航系统导航信息融合算法的仿真和实验

在此利用 MATLAB 进行 GPS/DR 综合导航系统导航信息融合的仿真。为了验证所设计的 GPS/DR 综合导航系统导航信息融合算法的有效性, 分别采用设计的融合算法和卡尔曼滤波技术来进行导航信息融合的仿真, 然后将两次的仿真结果进行比较。在仿真中采用的是真实的 GPS 数据。总共 2200 s 的 GPS 数据被用于此次的仿真。所设计的 GPS/DR 综合导航系统导航信息融合算法与卡尔曼滤波的融合结果相比较。图 2-20 和图 2-21 分别展示了 GPS/DR 导航数据融合结果的误差。从仿真结果可以看出, 针对所选的第一款民用单频 GPS 接收机数据特点所设计的 GPS/DR 综合导航系统导航信息融合算法可以提供比传统的卡尔曼滤波技术更好的导航信息融合结果。而且, 在某些特殊情况下, 由于一些突发的情况, 例如, GPS 卫星信号不稳定、周围出现高大建筑物等情况, 使得 GPS 的定位精度大幅度的下降甚至无法正常工作。在这种情况下卡尔曼滤波技术将无法提供很好的导航信息融合结果。而书中提出的基于 GPS 接收机数据特点所设计的导航信息融合算法可以通过自适应方式来改变 GPS 的可信度从而解决这一问题。当 GPS 的工作情况不好时可以将 GPS 的可信度调低, 这种情况下可以主要利用 DR 系统来提供导航信息。而当 GPS 恢复正常工作时, GPS 的可信度也随之恢复, GPS/DR 综合导航系统恢复正常工作状态。

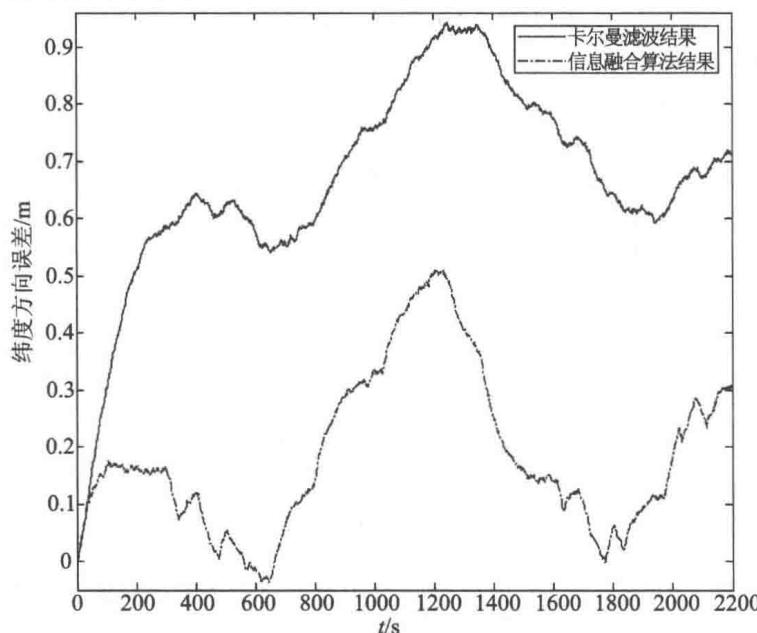


图 2-20 纬度方向设计的 GPS/DR 导航信息融合算法和卡尔曼滤波技术所得到的融合结果的误差

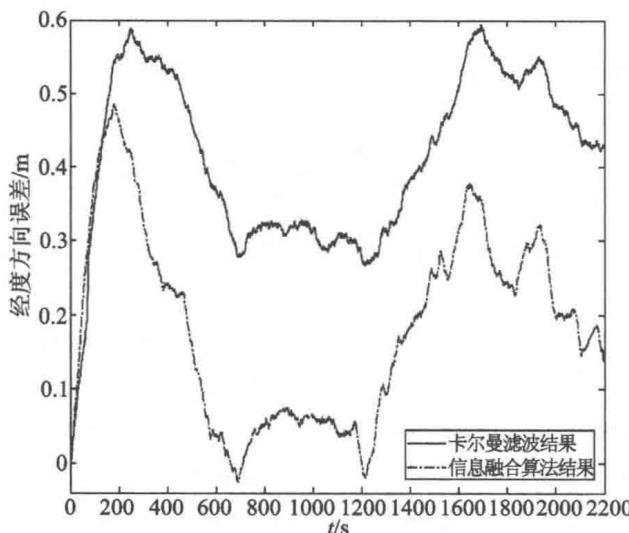


图 2-21 经度方向设计的 GPS/DR 导航信息融合算法和
卡尔曼滤波技术所得到的融合结果的误差

随后,利用实验室中的移动机器人“DU ROBO”对所设计的 GPS/DR 综合导航系统导航信息融合算法进行实验验证。该机器人拥有两个驱动轮和一个辅助轮。机器人上配备有两个轮式编码器、一个电子罗盘和一个单频 GPS 接收机作为导航传感器。实验中机器人的路径被设定为一个 $10\text{ m} \times 10\text{ m}$ 的正方形。机器人需要绕着该正方形运行 3 圈。图 2-22 展示了实验的结果(正方形路径底边处真实路径和参考路径已重合)。

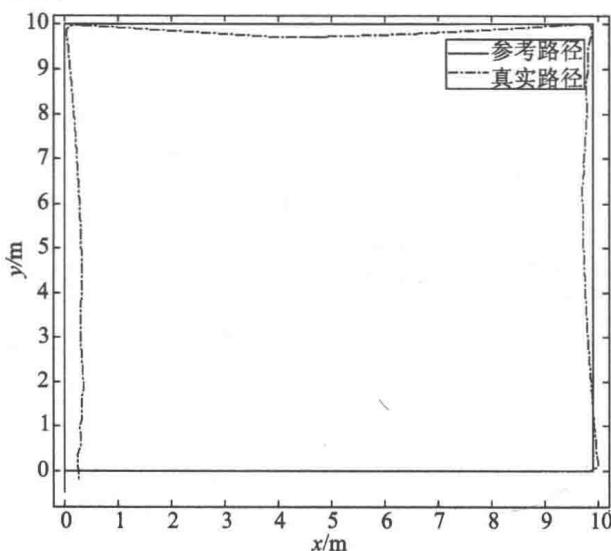


图 2-22 基于所选第一款民用单频 GPS 接收机数据特点的 GPS/DR
综合导航系统导航信息融合算法的实验结果

3) 基于第二款 GPS 接收机数据特点和神经网络的 GPS/DR 数据融合算法的设计

DR 系统可以提供短期精确的导航信息,然而 DR 系统的误差会随着时间的增加而累积。由于 GPS 与 DR 之间的互补的特点,可以利用 GPS/DR 综合导航系统为移动机器人提供长期稳定的、准确的导航信息。基于第二款民用单频 GPS 接收机数据特点和神经网络所设计的 GPS/DR 综合导航系统导航信息融合算法利用 DR 系统提供短期精确的导航信息,然后利用民用单频 GPS 接收机的输出进行长期的导航数据的修正。所选的第二款民用单频 GPS 接收机的数据特点为:该 GPS 接收机的误差变化频繁,但是相邻的输出数据有一定的关系,并且两个相邻的输出数据之间的差值不超过三倍的 GPS 的分辨率,对于纬度方向来说为 0.1854 m,对于经度方向来说为 0.1503 m。采用训练好的 BP 神经网络基于前三个采样时刻的 GPS 接收机的输出值和相应的 HDOP 值来对当前采样时刻 GPS 接收机的输出数据进行预测。图 2-23 展示了所设计的 GPS/DR 综合导航系统导航信息融合算法的简图。

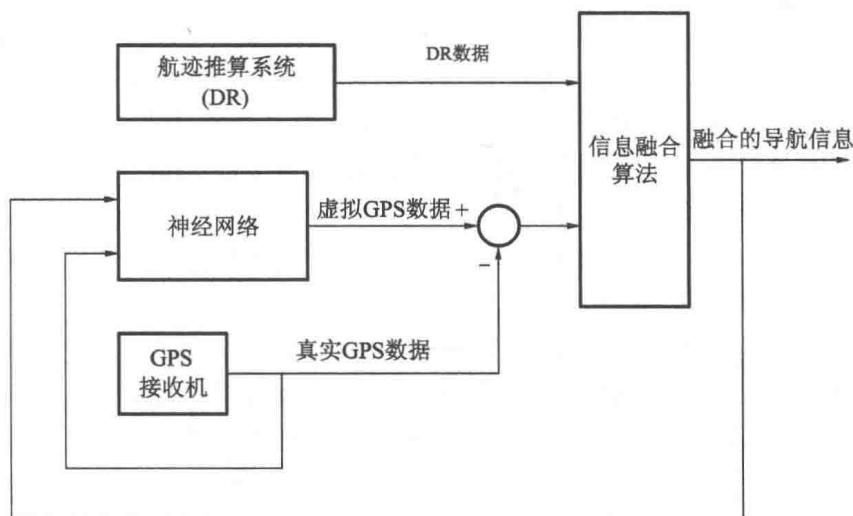


图 2-23 基于第二款民用 GPS 接收机数据特点和神经网络所设计的 GPS/DR 综合导航系统导航信息融合算法简图

基于所选第二款民用单频 GPS 接收机数据特点和神经网络所设计的 GPS/DR 综合导航系统导航信息融合算法如下。同上所述, x 用来表示纬度方向, y 用来表示经度方向。设置基于 DR 系统的移动机器人在 x 方向和 y 方向的坐标的可信度分别为 Q_x 和 Q_y ; 设置基于所选第二种民用单频 GPS 接收机的移动机器人在 x 方向和 y 方向的坐标的可信度分别为 R_x 和 R_y 。由于 DR 系统可以提供短时精确的导航信息,而民用单频 GPS 接收机的绝对误差很大,因此 Q_x 和 Q_y 的值要远

远大于 R_x 和 R_y 的值。

假设在移动机器人的出发点上有一个假想的民用单频 GPS 接收机, 该 GPS 接收机与固定在移动机器人上的所选的第二种民用单频 GPS 接收机的数据特点一致。利用一个训练好的如前所述的神经网络来预测固定在出发点上的假想 GPS 接收机的输出值。该神经网络的输出即是该假想 GPS 接收机的输出值。

①首先, 采集所选 GPS 接收机的数据进行神经网络的训练和测试, 以便得到能够很好预测假想 GPS 接收机输出数据的神经网络。

②在出发点上, 在移动机器人没有出发时采集三个采样时刻的 GPS 接收机的输出值: $GPS_x(0), GPS_x(1), GPS_x(2), GPS_y(0), GPS_y(1)$ 与 $GPS_y(2)$ 和相应的 HDOP 值。其中 $GPS_x(0), GPS_x(1), GPS_x(2), GPS_y(0), GPS_y(1)$ 和 $GPS_y(2)$ 分别是 GPS 接收机在时刻 $t=0, 1, 2$ s 时的纬度方向和经度方向的输出值。假设固定在出发点上的假想 GPS 接收机在时刻 $t=0, 1, 2$ s 时有与该真实 GPS 接收机同样的输出值, 也就是说 $GPS_{I_x}(0)=GPS_x(0), GPS_{I_x}(1)=GPS_x(1), GPS_{I_x}(2)=GPS_x(2), GPS_{I_y}(0)=GPS_y(0), GPS_{I_y}(1)=GPS_y(1)$ 和 $GPS_{I_y}(2)=GPS_y(2)$ 。其中 $GPS_{I_x}(0), GPS_{I_x}(1), GPS_{I_x}(2), GPS_{I_y}(0), GPS_{I_y}(1)$ 和 $GPS_{I_y}(2)$ 分别是假想 GPS 接收机在时刻 $t=0, 1, 2$ s 时纬度方向和经度方向的输出值。同时也可得到在这三个采样时刻的 HDOP 值: HDOP(1), HDOP(2) 和 HDOP(3)。

③运行移动机器人。由于所选的民用单频 GPS 接收机每 1 s 输出一次导航数据, 因此 GPS/DR 导航数据融合算法的时间间隔选为 1 s。在时刻 $t=3$ s 通过 DR 系统可以得到基于 DR 系统的移动机器人坐标 $x_{DR}(3)$ 和 $y_{DR}(3)$ 。其中 $x_{DR}(3)$ 和 $y_{DR}(3)$ 是基于 DR 系统的移动机器人在纬度方向和经度方向上的坐标值。设置 $x_{DR}(3)$ 和 $y_{DR}(3)$ 的可信度分别为 Q_x 和 Q_y 。

④利用三个采样周期的假想 GPS 接收机的输出和相应的 HDOP 值以及神经网络来预估当前时刻的假想 GPS 接收机的输出。也就是说将 $GPS_{I_x}(0), GPS_{I_x}(1), GPS_{I_x}(2), GPS_{I_y}(0), GPS_{I_y}(1), GPS_{I_y}(2), HDOP(1), HDOP(2)$ 和 $HDOP(3)$ 通过事先训练好的神经网络输出为当前时刻假想 GPS 接收机的输出值 $GPS_{I_x}(3)$ 和 $GPS_{I_y}(3)$ 。其中符号“~”表示的是假想 GPS 接收机输出的预估值。

⑤在时刻 $t=3$ s 时, 通过真实的 GPS 接收机可以得到当前时刻的 GPS 输出值: $GPS_x(3)$ 和 $GPS_y(3)$ 。

⑥利用真实的 GPS 接收机输出值和固定在出发点上的假想 GPS 接收机输出值的差值来预估在时刻 $t=3$ s 时基于 GPS 的移动机器人坐标值: $x_{GPS}(3)=GPS_x(3)-GPS_{I_x}(3)$ 和 $y_{GPS}(3)=GPS_y(3)-GPS_{I_y}(3)$ 。其中 $x_{GPS}(3)$ 和 $y_{GPS}(3)$ 分别是基于 GPS 的移动机器人在纬度方向和经度方向上的坐标值。

⑦设置基于 GPS 的移动机器人坐标值 $x_{GPS}(3)$ 和 $y_{GPS}(3)$ 的可信度 R_x 和 R_y 。由于 DR 系统可以提供短时准确的导航信息, 因此在一个采样周期内(也就是 1 s

内),DR 系统的导航数据是非常精确的。因此可以利用 DR 系统在当前时刻的导航数据来设置基于 GPS 的移动机器人坐标值的可信度。由于所选的第二款民用单频 GPS 接收机相邻两个采样时刻的数据变化不大,不超过三倍的 GPS 接收机的分辨率,也就是说纬度方向 0.1854 m 和经度方向 0.1503 m,因此可以按照如下方法对基于 GPS 的移动机器人的位置坐标可信度进行设置。对于纬度方向来说:如果 $|x_{\text{GPS}}(3) - x_{\text{DR}}(3)| \leq 0.1854 \text{ m}$,则可以设置 $x_{\text{GPS}}(3)$ 的可信度为 R_{x1} ;如果 $|x_{\text{GPS}}(3) - x_{\text{DR}}(3)| \leq 0.1854 \times 2 \text{ m}$,则可以设置 $x_{\text{GPS}}(3)$ 的可信度为 R_{x2} ;如果 $|x_{\text{GPS}}(3) - x_{\text{DR}}(3)| \leq 0.1854 \times 3 \text{ m}$,则可以设置 $x_{\text{GPS}}(3)$ 的可信度为 R_{x3} ;如果 $|x_{\text{GPS}}(3) - x_{\text{DR}}(3)| \geq 0.1854 \times 3 \text{ m}$,这种情况的发生说明 GPS 的输出发生了大幅度的改变,这时 GPS 接收机工作在一种不良的工作状况下从而导致 GPS 接收机的导航精度下降很大,在这种情况下 $x_{\text{GPS}}(3)$ 的可信度设置为 R_{x4} 。同样的方法可以用来设置基于 GPS 的移动机器人经度方向的坐标值的可信度。类似于设置 $x_{\text{GPS}}(3)$ 的可信度的方法,基于 $|y_{\text{GPS}}(3) - y_{\text{DR}}(3)|$ 的值可以将 $y_{\text{GPS}}(3)$ 的可信度分别设置为 R_{y1} 、 R_{y2} 、 R_{y3} 和 R_{y4} 。由于 DR 系统可以提供短期(1 s)精确的导航信息,其准确度比所选民用单频 GPS 接收机的精度要高得多,因此 Q_x 和 Q_y 要比 R_{xi} 和 R_{yi} ($i=1,2,3,4$) 大得多。并且有如下的关系。

$$\begin{aligned} R_{x1} &> R_{x2} > R_{x3} \gg R_{x4} \\ R_{y1} &> R_{y2} > R_{y3} \gg R_{y4} \end{aligned} \quad (2-31)$$

⑧在 $t=3$ s 时刻, GPS/DR 综合导航系统导航信息融合的结果,也就是移动机器人导航信息融合后的坐标值可以由下式给出。

$$x(3) = \frac{Q_x}{Q_x + R_x} x_{\text{DR}}(3) + \frac{R_x}{Q_x + R_x} x_{\text{GPS}}(3) \quad (2-32)$$

$$y(3) = \frac{Q_y}{Q_y + R_y} y_{\text{DR}}(3) + \frac{R_y}{Q_y + R_y} y_{\text{GPS}}(3) \quad (2-33)$$

其中: $x(3)$ 和 $y(3)$ 是融合后的移动机器人的坐标值。移动机器人的方向可以由电子罗盘来提供。

⑨在当前时刻固定在出发点的假想 GPS 接收机的输出值可以由下式进行计算。

$$\begin{aligned} \text{GPS}_{I_x}(3) &= \text{GPS}_x(3) - x(3) \\ \text{GPS}_{I_y}(3) &= \text{GPS}_y(3) - y(3) \end{aligned} \quad (2-34)$$

⑩在时刻 $t=k$ s ($k=4,5,\dots$),基于 DR 系统可以得到当前时刻基于 DR 系统的移动机器人的坐标值 $x_{\text{DR}}(k)$ 和 $y_{\text{DR}}(k)$ 。设置坐标值 $x_{\text{DR}}(k)$ 和 $y_{\text{DR}}(k)$ 的可信度分别为 Q_x 和 Q_y 。

⑪将前三个采样时刻的假想 GPS 接收机的输出和相应的真实 GPS 接收机的 HDOP 值通过神经网络去预估当前时刻假想 GPS 接收机的输出值: $\text{GPS}_{I_x}(k)$ 和

$\text{GPS}_{L\hat{x}}(k)$ 。

⑫同时,真实 GPS 接收机的输出为 $\text{GPS}_x(k)$ 和 $\text{GPS}_y(k)$ 。

⑬基于当前时刻真实 GPS 接收机的输出和假想 GPS 接收机的输出的差值来计算基于 GPS 的移动机器人的坐标值: $x_{\text{GPS}}(k) = \text{GPS}_x(k) - \text{GPS}_{L\hat{x}}(k)$ 和 $y_{\text{GPS}}(k) = \text{GPS}_y(k) - \text{GPS}_{L\hat{y}}(k)$ 。

⑭类似于步骤⑦,可以进行 $x_{\text{GPS}}(k)$ 和 $y_{\text{GPS}}(k)$ 的可信度的设置。

⑮在时刻 $t=k$ s 时,利用下式进行 GPS/DR 综合导航系统导航信息的融合解算,也就是解算移动机器人导航信息融合后的坐标值。

$$x(k) = \frac{Q_x}{Q_x + R_x} x_{\text{DR}}(k) + \frac{R_x}{Q_x + R_x} x_{\text{GPS}}(k) \quad (2-35)$$

$$y(k) = \frac{Q_y}{Q_y + R_y} y_{\text{DR}}(k) + \frac{R_y}{Q_y + R_y} y_{\text{GPS}}(k) \quad (2-36)$$

⑯当前时刻固定在出发点的假想 GPS 接收机的输出为

$$\begin{aligned} \text{GPS}_{L\hat{x}}(k) &= \text{GPS}_x(k) - x(k) \\ \text{GPS}_{L\hat{y}}(k) &= \text{GPS}_y(k) - y(k) \end{aligned} \quad (2-37)$$

4) 基于第二种 GPS 接收机导航数据特点和神经网络的信息融合算法的仿真

在本书中利用 MATLAB 对所设计的 GPS/DR 综合导航系统导航信息融合算法进行了仿真。仿真中使用了真实的 GPS 数据。如前所述,采样训练好的 BP 神经网络对当前假想 GPS 接收机的输出值进行了预估。首先采集所选的第二款 GPS 接收机的数据作为训练和测试神经网络的数据。利用 MATLAB 的神经网络工具箱来训练和测试该神经网络。在得到了训练好的神经网络后,采用真实的 GPS 数据进行 GPS/DR 导航信息融合算法的仿真。为了验证所提出的 GPS/DR 导航信息融合算法,采用该算法和卡尔曼滤波技术分别进行仿真,并将两者的仿真结果进行比较。图 2-24 和图 2-25 展示了采用所设计的 GPS/DR 导航信息融合算法和卡尔曼滤波技术所得到的导航信息融合仿真结果。从图中结果可以看出来,针对所选用的第二款 GPS 接收机来说所设计的 GPS/DR 导航信息融合算法可以提供比卡尔曼滤波技术更好的信息融合结果。而且,在某些特殊情况下,由于一些突发的情况(例如 GPS 卫星信号不稳定,周围出现高大建筑物等情况),GPS 的定位精度会出现大幅度的下降甚至无法正常工作。在这种情况下卡尔曼滤波技术将无法提供很好的导航信息融合结果,而书中提出的基于 GPS 接收机数据特点所设计的导航信息融合算法可以通过自适应方式来改变 GPS 的可信度从而解决这一问题。当 GPS 的工作情况不好时可以将 GPS 的可信度调低,这种情况下可以主要利用 DR 系统来提供导航信息。而当 GPS 恢复正常工作时,GPS 的可信度也随之恢复,GPS/DR 综合导航系统恢复正常工作状态,从而提高了 GPS/DR 综合导航系统工作的可靠性。

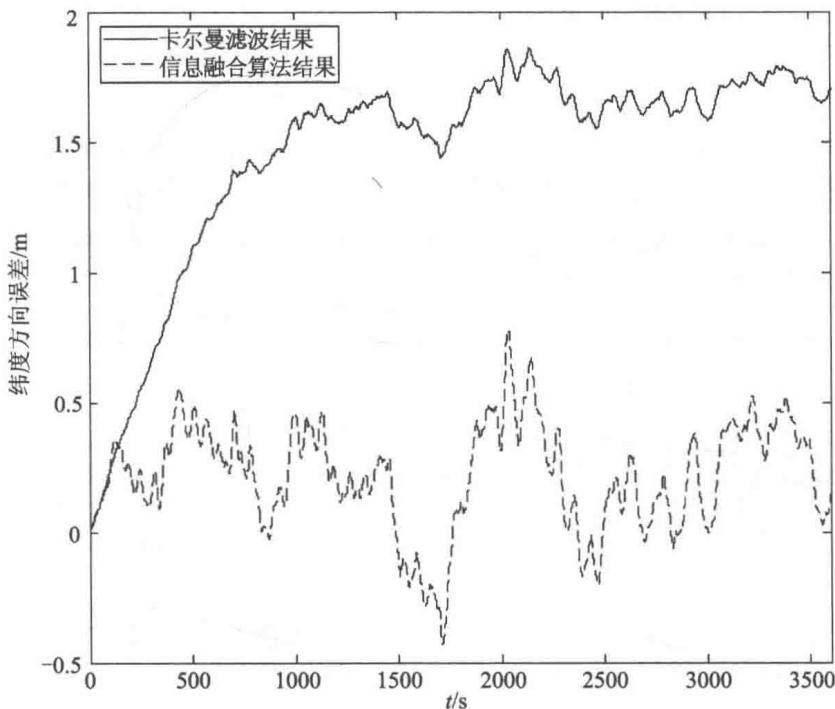


图 2-24 基于所选第二款 GPS 接收机和神经网络所设计的 GPS/DR 综合导航系统导航信息融合算法和卡尔曼滤波技术的导航信息融合结果(纬度方向上的误差)

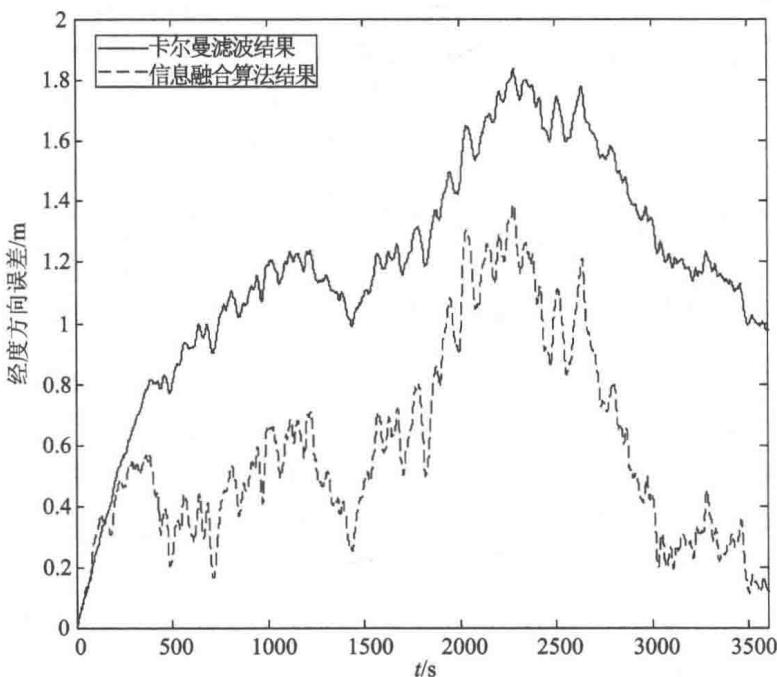


图 2-25 基于所选第二款 GPS 接收机和神经网络所设计的 GPS/DR 综合导航系统导航信息融合算法和卡尔曼滤波技术的导航信息融合结果(经度方向上的误差)

5) 基于第二款 GPS 接收机数据特点的 GPS/DR 数据融合算法的设计

前面所述的 GPS/DR 综合导航系统导航信息融合算法是基于所选第二款民用单频 GPS 接收机的数据特点并利用神经网络的预测能力来进行设计的。下面所述的 GPS/DR 综合导航系统导航信息融合算法也是基于所选第二款民用单频 GPS 接收机的数据特点而设计的,只是不采用神经网络,而是利用窗口技术来实现的。所选第二款 GPS 接收机的数据如图 2-26 所示。

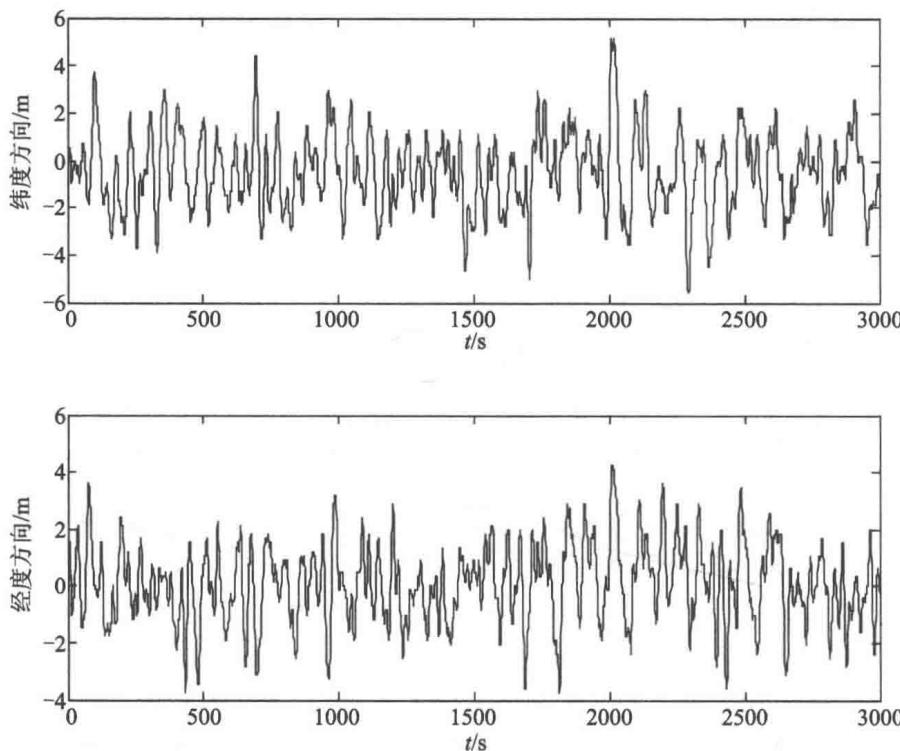


图 2-26 所选第二款 GPS 接收机减去平均值后的导航数据

从图 2-26 可以看出所选第二款民用单频 GPS 接收机的导航数据变化较快且数据之间的关系不明显,因此无法直接用于 GPS/DR 导航信息融合算法的设计,需要在此数据的基础上进行进一步处理。考虑用一个固定长度包含若干个 GPS 采样周期的时间窗口,在此窗口中的 GPS 接收机的数据的平均值可以作为导航的数据。然后移动该时间窗口至下一个采样时刻,得到下一个 GPS 数据的平均值。关于窗口的长度需要做多次的实验来确定。经过多次的实验发现 100 个 GPS 采样时刻(即 100 s)的时间窗口就可以得到理想的结果。经过时间窗口处理完的数据如图 2-27 所示。

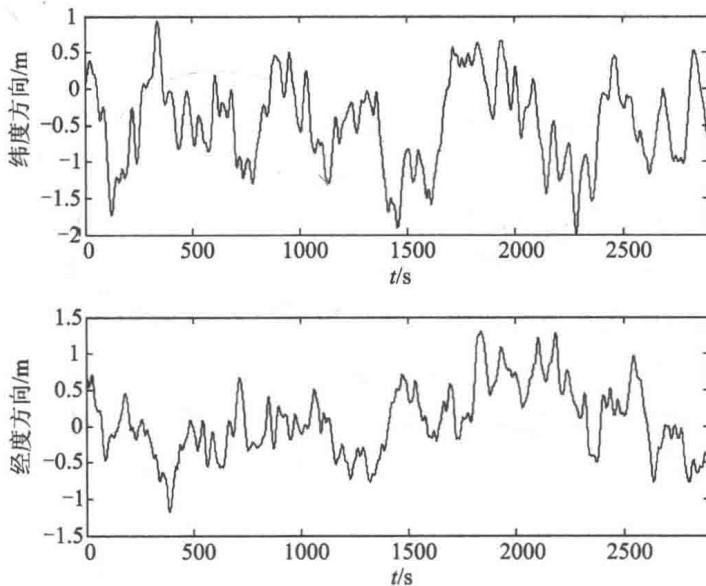


图 2-27 经过时间窗口处理后的 GPS 接收机的数据

图 2-27 所显示的数据要比图 2-26 中的数据好得多。但是从该数据中仍然不容易得出很有用的信息,因此如前所述进行进一步的处理。也就是说,在图 2-27 的数据中,用当前数据减去前一个采样时刻的数据,将得到的处理后的数据显示在图 2-28 中。图 2-28 中的数据是相邻采样时刻的图 2-27 中的数据的差值,从图 2-28 中可以看出图 2-27 相邻采样时刻的数据的差值很小,可以利用这样的关系来进行 GPS/DR 综合导航系统导航信息融合算法的设计。

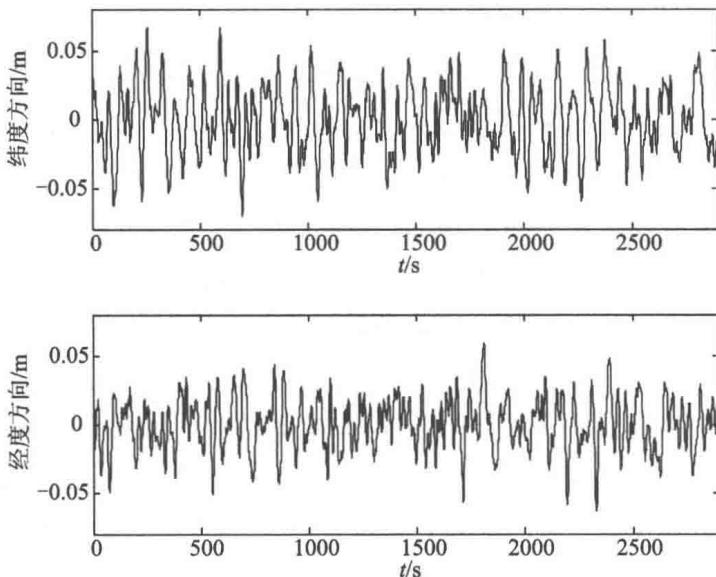


图 2-28 相邻采样时刻的处理后的 GPS 数据之间的差值

所选第二款 GPS 接收机的数据特点如下:相邻采样时刻,利用时间窗口技术处理后的 GPS 数据之间的差距非常小,在纬度方向和经度方向均小于 0.8 m。DR 系统可以提供短时准确的导航信息,但是其误差会随着时间而不断累积。在本 GPS/DR 综合导航系统中,DR 系统用于提供短期准确的导航数据,然后利用 GPS 进行长期导航数据的调整。同上所述, x 用来表示纬度方向, y 用来表示经度方向。设置基于 DR 系统的移动机器人在 x 方向和 y 方向的坐标的可信度分别为 Q_x 和 Q_y ;设置基于所选第二种民用单频 GPS 接收机的移动机器人在 x 方向和 y 方向的坐标的可信度分别为 R_x 和 R_y 。由于 DR 系统可以提供短时准确的导航信息,而民用单频 GPS 接收机的绝对误差很大,因此 Q_x 和 Q_y 的值要远远大于 R_x 和 R_y 的值。所设计的 GPS/DR 综合导航系统导航信息融合算法如下。

假设在出发点上固定一个假想的 GPS 接收机,该 GPS 接收机与固定在移动机器人上的真实 GPS 接收机有相同的数据特点。

①在出发点上移动机器人没出发时,当所选 GPS 接收机的输出稳定后,采集 GPS 接收机 100 个采样周期的数据,然后计算其平均值: $\bar{GPS}_x(0)$ 和 $\bar{GPS}_y(0)$ 。其中 $\bar{GPS}_x(0)$ 和 $\bar{GPS}_y(0)$ 是在时刻 $t=0$ s 时所选 GPS 接收机 100 个采样周期的数据平均值。假设在出发点上,固定在出发点上的假想 GPS 接收机有着和真实 GPS 接收机一样的输出: $GPS_{L_x}(0) = \bar{GPS}_x(0)$ 和 $GPS_{L_y}(0) = \bar{GPS}_y(0)$ 。其中 $\bar{GPS}_{L_x}(0)$ 和 $\bar{GPS}_{L_y}(0)$ 是在时刻 $t=0$ s 时假想 GPS 接收机 100 个采样周期的数据的平均值。

②运行移动机器人。由于 GPS 接收机的输出频率是 1 Hz,因此所设计的 GPS/DR 综合导航系统导航信息融合算法的周期是 1 s。基于 DR 系统的移动机器人纬度方向和经度方向的坐标值分别为 $x_{DR}(1)$ 和 $y_{DR}(1)$ 。设置 $x_{DR}(1)$ 和 $y_{DR}(1)$ 的可信度分别为 Q_x 和 Q_y 。

③在时刻 $t=1$ s 时,采集 GPS 的数据 $GPS_x(1)$ 和 $GPS_y(1)$ 。其中 $GPS_x(1)$ 和 $GPS_y(1)$ 分别是时刻 $t=1$ s 时 GPS 接收机纬度方向和经度方向的输出值。由于 DR 系统可以提供短期准确的导航数据,因此将 $GPS_x(1) - x_{DR}(1)$ 和 $GPS_y(1) - y_{DR}(1)$ 投入新的时间窗口并求出新时间窗口中的数据平均值: $\bar{GPS}_x(1)$ 和 $\bar{GPS}_y(1)$ 。

④经过窗口技术处理后的所选第二款民用单频 GPS 接收机的数据,由于相邻时刻的数据相差很小,因此可以预测在时刻 $t=1$ s 时固定在出发点上的假想 GPS 接收机的输出为 $GPS_{L_x}(1) = \bar{GPS}_{L_x}(0)$ 和 $GPS_{L_y}(1) = \bar{GPS}_{L_y}(0)$ 。其中“~”表示

为假想 GPS 接收机输出的平均值。

⑤在时刻 $t=1$ s 时,可以预测基于 GPS 的移动机器人纬度方向和经度方向的坐标分别为 $x_{\text{GPS}}(1) = \text{GPS}_x(1) + x_{\text{DR}}(1) - \text{GPS}_{L\bar{x}}(1)$ 和 $y_{\text{GPS}}(1) = \text{GPS}_y(1) + y_{\text{DR}}(1) - \text{GPS}_{L\bar{y}}(1)$ 。

⑥设置 $x_{\text{GPS}}(1)$ 和 $y_{\text{GPS}}(1)$ 的可信度分别为 R_x 和 R_y 。由于 DR 系统可以提供短期准确的导航信息,因此可以利用在时刻 $t=1$ s 时基于 DR 系统的移动机器人纬度方向和经度方向的坐标值 $x_{\text{DR}}(1)$ 和 $y_{\text{DR}}(1)$ 来设置 $x_{\text{GPS}}(1)$ 和 $y_{\text{GPS}}(1)$ 的可信度 R_x 和 R_y 。对于纬度方向来说:如果 $|x_{\text{GPS}}(1) - x_{\text{DR}}(1)| \leq 0.08$ m, 则 $x_{\text{GPS}}(1)$ 的可信度为 R_{x1} ; 如果 $|x_{\text{GPS}}(1) - x_{\text{DR}}(1)| > 0.08$ m, 这说明由于发生了一些突发情况使得 GPS 的数据跳动比较大, 这时 GPS 工作在一种不良的工作状况下从而导致 GPS 接收机的导航精度下降很大, 在这种情况下 $x_{\text{GPS}}(1)$ 的可信度设置为 R_{x2} 。对于经度方向来说:如果 $|y_{\text{GPS}}(1) - y_{\text{DR}}(1)| \leq 0.08$ m, 则设置 $y_{\text{GPS}}(1)$ 的可信度为 R_{y1} ; 如果 $|y_{\text{GPS}}(1) - y_{\text{DR}}(1)| > 0.08$ m, 则设置 $y_{\text{GPS}}(1)$ 的可信度为 R_{y2} , 并且有如公式(2-38)所示的关系存在。

$$R_{x1} > R_{x2}$$

$$R_{y1} > R_{y2} \quad (2-38)$$

⑦移动机器人的方向信息由电子罗盘来提供。在时刻 $t=1$ s 时融合后的 GPS/DR 导航信息可以由公式(2-39)和公式(2-40)来解算。

$$x(1) = \frac{Q_x}{Q_x + R_x} x_{\text{DR}}(1) + \frac{R_x}{Q_x + R_x} x_{\text{GPS}}(1) \quad (2-39)$$

$$y(1) = \frac{Q_y}{Q_y + R_y} y_{\text{DR}}(1) + \frac{R_y}{Q_y + R_y} y_{\text{GPS}}(1) \quad (2-40)$$

⑧在时刻 $t=1$ s 时, 固定在出发点上的假想 GPS 接收机的输出可以由公式(2-41)给出。将 $\text{GPS}_{L\bar{x}}(1)$ 和 $\text{GPS}_{L\bar{y}}(1)$ 投入到新的时间窗口中并计算新时间窗口中数据的平均值: $\text{GPS}_{L\bar{x}}(1)$ 和 $\text{GPS}_{L\bar{y}}(1)$ 。

$$\text{GPS}_{L\bar{x}}(1) = \text{GPS}_x(1) - x(1)$$

$$\text{GPS}_{L\bar{y}}(1) = \text{GPS}_y(1) - y(1) \quad (2-41)$$

⑨在时刻 $t=k$ s ($k=2, 3, \dots$), 基于 DR 系统的移动机器人纬度方向和经度方向的坐标值为 $x_{\text{DR}}(k)$ 和 $y_{\text{DR}}(k)$, 设置 $x_{\text{DR}}(k)$ 和 $y_{\text{DR}}(k)$ 的可信度分别为 Q_x 和 Q_y 。

⑩采集 GPS 接收机的数据得到 GPS 在纬度方向和经度方向的输出值 $\text{GPS}_x(k)$ 和 $\text{GPS}_y(k)$ 。将 $\text{GPS}_x(k) - x_{\text{DR}}(k)$ 和 $\text{GPS}_y(k) - y_{\text{DR}}(k)$ 投入到新的时间窗口中并计算新的时间窗口中数据的平均值 $\text{GPS}_{\bar{x}}(k)$ 和 $\text{GPS}_{\bar{y}}(k)$ 。

⑪预估当前时刻在时间窗口中假想 GPS 接收机输出的平均值: $GPS_{\bar{x}}(k) = GPS_{\bar{x}}(k-1)$ 和 $GPS_{\bar{y}}(k) = GPS_{\bar{y}}(k-1)$ 。

⑫预估基于 GPS 的移动机器人纬度方向和经度方向的坐标值: $x_{GPS}(k) = GPS_x(k) + x_{DR}(k) - GPS_{\bar{x}}(k)$ 和 $y_{GPS}(k) = GPS_y(k) + y_{DR}(k) - GPS_{\bar{y}}(k)$ 。

⑬类似于步骤⑥来进行 $x_{GPS}(k)$ 和 $y_{GPS}(k)$ 的可信度的设置。

⑭利用公式(2-42)和公式(2-43)来进行在时刻 $t=k$ s 的 GPS/DR 综合导航系统导航信息融合结果的解算。

$$x(k) = \frac{Q_x}{Q_x + R_x} x_{DR}(k) + \frac{R_x}{Q_x + R_x} x_{GPS}(k) \quad (2-42)$$

$$y(k) = \frac{Q_y}{Q_y + R_y} y_{DR}(k) + \frac{R_y}{Q_y + R_y} y_{GPS}(k) \quad (2-43)$$

⑮当前时刻假想 GPS 接收机的输出可以由公式(2-44)给出。然后将 $GPS_{\bar{x}}(k)$ 和 $GPS_{\bar{y}}(k)$ 投入到新的时间窗口中并计算其中数据的平均值: $GPS_{\bar{x}}(k)$ 和 $GPS_{\bar{y}}(k)$ 。

$$\begin{aligned} GPS_{\bar{x}}(k) &= GPS_x(k) - x(k) \\ GPS_{\bar{y}}(k) &= GPS_y(k) - y(k) \end{aligned} \quad (2-44)$$

6) 基于第二种 GPS 接收机导航数据特点和时间窗口技术的信息融合算法的仿真

在仿真中采用真实的 GPS 数据进行所设计的 GPS/DR 综合导航系统导航信息融合算法的仿真验证。仿真中采用总共 4000 个采样周期的 GPS 数据。为了更好地验证所设计的 GPS/DR 综合导航系统导航信息融合算法, 将仿真结果与卡尔曼滤波技术得到的融合结果相比较。纬度方向和经度方向的导航仿真结果分别显示在图 2-29 和图 2-30 中。从结果可以看出对于所选的第二款 GPS 接收机来说所设计的 GPS/DR 综合导航系统导航信息融合算法可以提供比卡尔曼滤波技术更好的导航结果。而且, 在某些特殊情况下, 由于一些突发的情况(例如 GPS 卫星信号不稳定, 周围出现高大建筑物等情况), GPS 的定位精度会出现大幅度的下降甚至无法正常工作。在这种情况下卡尔曼滤波技术将无法提供很好的导航信息融合结果。而书中提出的基于 GPS 接收机数据特点所设计的导航信息融合算法可以通过自适应方式来改变 GPS 的可信度从而解决这一问题。当 GPS 的工作情况不好时可以将 GPS 的可信度调低, 这种情况下可以主要利用 DR 系统来提供导航信息, 而当 GPS 恢复正常工作时, GPS 的可信度也随之恢复, GPS/DR 综合导航系统恢复正常工作状态, 从而提高了 GPS/DR 综合导航系统工作的可靠性。

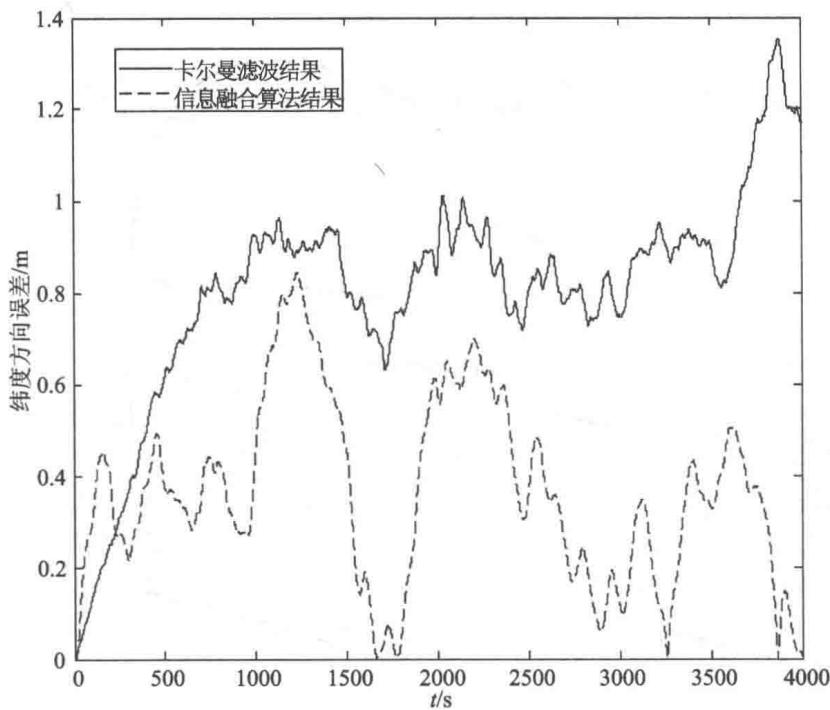


图 2-29 GPS/DR 导航信息融合的仿真结果(纬度方向)

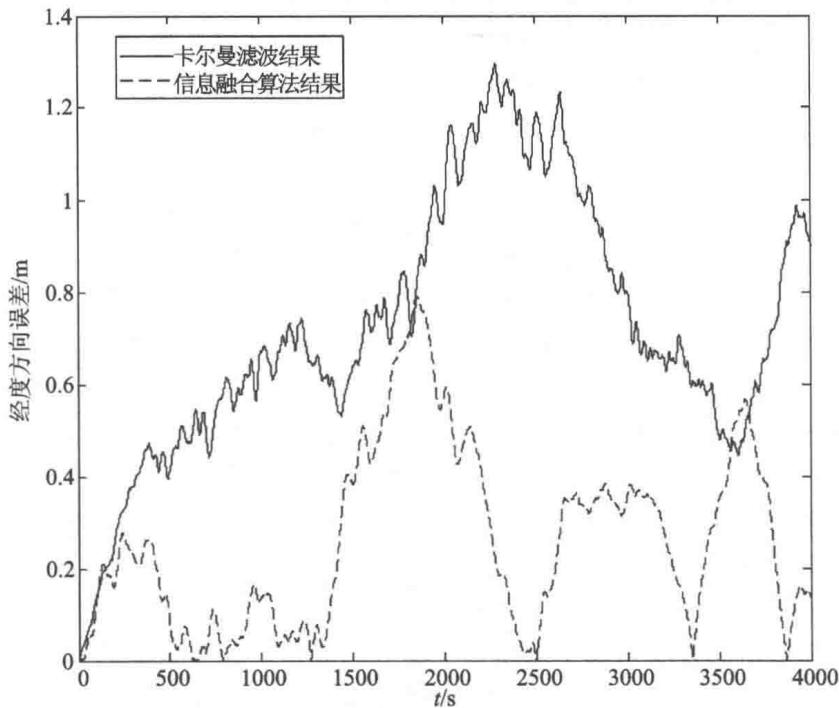


图 2-30 GPS/DR 导航信息融合的仿真结果(经度方向)

第3章 移动机器人的路径规划

3.1 A* 算法

A* (A-Star) 算法是一种静态路网中求解最短路径最有效的方法^[83]。公式表示为: $f(n) = g(n) + h(n)$ 。其中, $f(n)$ 是从初始点经由节点 n 到目标点的估价函数, $g(n)$ 是在状态空间中从初始节点到 n 节点的实际代价, $h(n)$ 是从 n 到目标节点最佳路径的估计代价。保证找到最短路径的(最优解的)条件, 关键在于估价函数 $h(n)$ 的选取: 估价值 $h(n) \leq n$ 到目标节点的距离实际值, 这种情况下, 搜索的点数多, 搜索范围大, 效率低, 但能得到最优解。如果估价值 > 实际值, 搜索的点数少, 搜索范围小, 效率高, 但不能保证得到最优解。下面通过一个简单的例子来说明 A* 算法。

关于 A* 算法的流程图见图 3-1。图 3-2 显示了本例中使用的简单的栅格地图。图 3-2 中, “S”指的是起始点, “E”指的是目标点, “O”指的是障碍物。首先要注意到的事情是, 搜索的区域被分割成许多正方形的栅格。每个栅格的状态分别被标记为“可航行区域”和“不可航行区域”。找到了通过哪些栅格区域能够从起始点“S”到达目标点“E”也就找到了一条可行的路径。这里, 每个栅格的中心点被称为“节点”。在获得了关于环境的栅格地图后, 下一步就是找到一条满足条件和限制的最优的路径。路径搜索要从起始点“S”开始, 然后搜索起始点“S”的各个临近点。依次搜索下去, 直到搜索到目标点“E”为止。

通过 A* 算法进行路径规划的步骤如下所示。

(1) 在初始时, 将起始点“S”加入到栅格的开放列表当中。这个开放列表中的栅格就是要被搜索的栅格。

(2) 找到与起始点“S”临近的除了包含障碍物的所有点, 将这些点加入到栅格的开放列表当中。对于这些栅格, 将起始点“S”标记为它们的“父点”。

(3) 将起始点“S”从栅格开放列表中剔除, 然后将其加入到栅格的关闭列表中。关闭列表中的栅格在以后的搜索中不会再被考虑。在图 3-3 中显示了前三个步骤的过程。在每个起始点“S”的临近栅格中有一个指针, 这个指针指向它的“父点”。

然后, 在栅格的开放列表中选择一个临近的栅格。拥有最低的 F 值的栅格将被选中。确定哪些栅格能够被采用的关键是由公式 $F = G + H$ 来决定的。公式中, F 是从初始点“S”经由当前节点到达目标点“E”的估价函数, G 为从初始点“S”沿着所产生的路径到达给定栅格的移动代价, H 为从当前给定栅格移动到终点

“E”的移动代价的估计值。这通常被称为启发式算法。

通过在栅格开放列表中不断重复地搜索拥有最低 F 值的栅格的过程就可以产生期望的最优路径。在这个例子中,对于参数 G 来说,每个水平或竖直的临近点被赋值为 10,每个对角线临近点被赋值为 14。当然,还可以用其他数值来赋值,这取决于对最优路径的具体要求。例如:是否局限于路径最短,或者同时还要考虑转过的角度最小,是否与障碍物保持一定距离等要求。

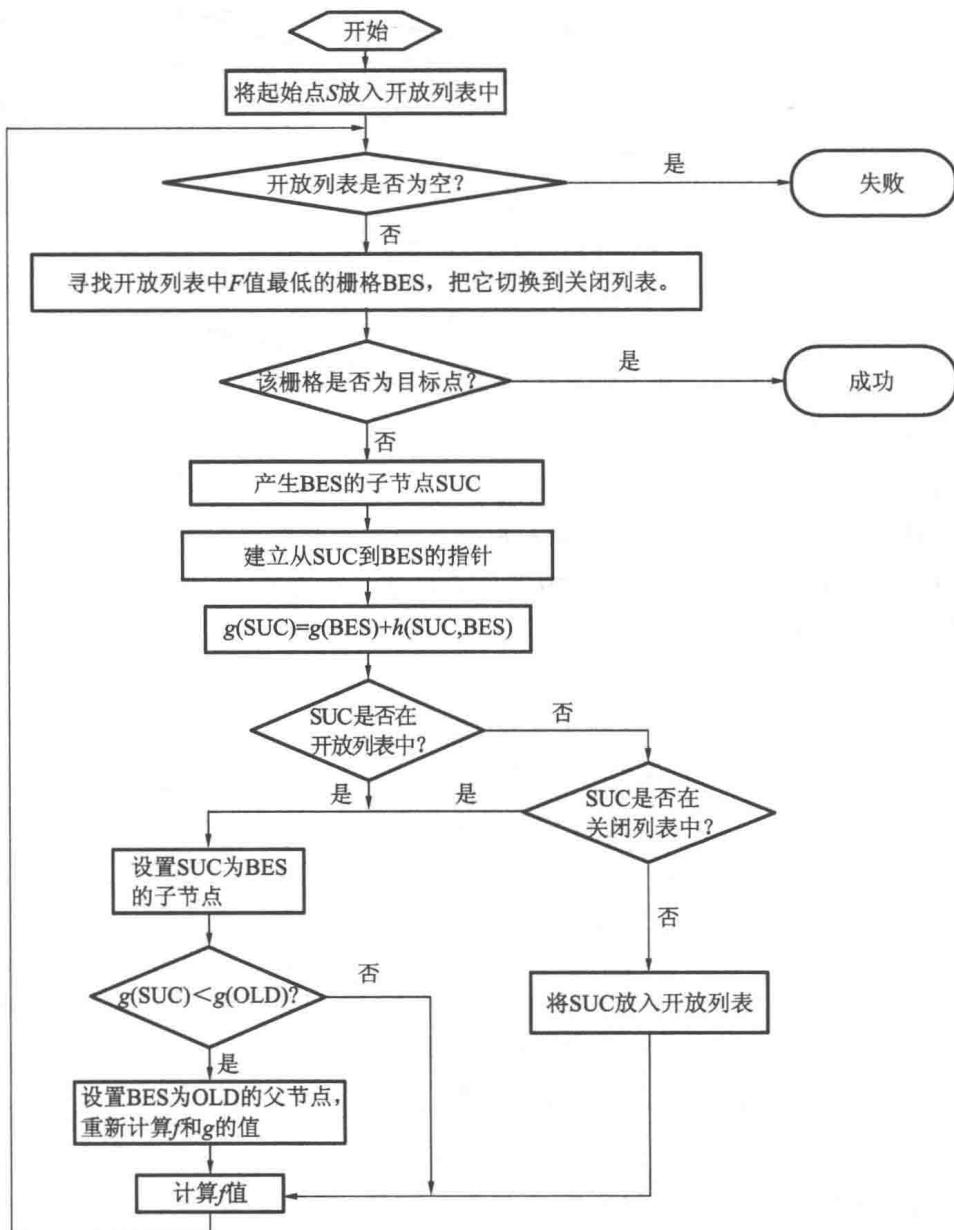


图 3-1 A* 算法流程图

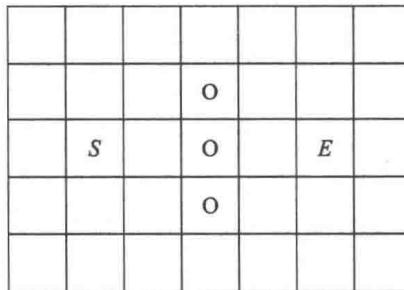


图 3-2 用于路径规划例子的简单的栅格地图

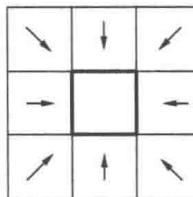
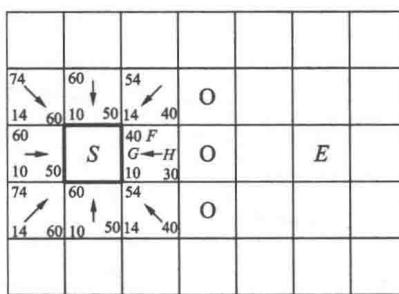


图 3-3 一个“父点”和它的临近点

参数 H 可以用“Manhattan”方法来估算。估算的方法为：从当前点到目标点的距离估算全部采用水平移动和竖直移动，而不采用对角线移动的方式来估算。在估算中，忽略所有路径上的障碍物。也就是说所有的障碍物都视为和可通过路径一样可以通过。当然也可以采用其他的方法来估算参数 H ，从而使得所产生的 A* 算法适合特定的应用。

第一步搜索的结果显示在图 3-4 中。

图 3-4 计算 F 值、 G 值和 H 值的方法

在栅格开放列表中拥有最低 F 值的栅格被选中，对被选中的栅格做如下操作。

(4) 将被选中的栅格从栅格开放列表中剔除，然后将其加入到栅格关闭列表中。

(5) 搜索该栅格的所有临近点,除了那些已经在栅格关闭列表中的或者被标记为不可走的栅格。然后将这些没有在栅格开放列表中的临近栅格加入到栅格开放列表中。将被选中的栅格标记为这些新加入到开放列表中的栅格的“父点”。

(6) 如果所选中的栅格的临近栅格已经存在于栅格开放列表中的话,则要判断是否当前路径为更好的路径。换句话说,就是看通过当前选择的栅格去达到那里的话,这个栅格的 G 值是否更低,如果不是的话,则对于原来就存在于栅格开放列表中的这些栅格不需要做任何的改动;另一方面,如果所得到的新的路径的 G 值更低的话,则将这些栅格的“父点”更改为当前选中的栅格。最终,重新计算这个栅格的 F 值和 G 值。图 3-5 显示了这个步骤的过程。

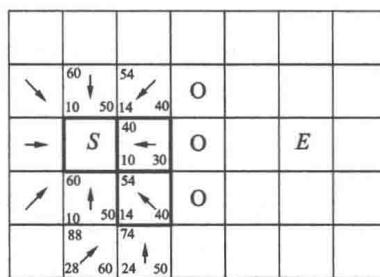


图 3-5 保持搜索路径的框图

重复如前所述的过程直到将目标点“E”加入到栅格关闭列表当中为止,见图 3-6。

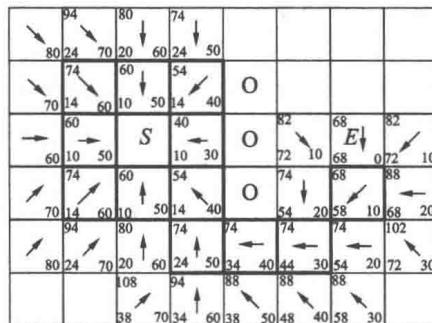


图 3-6 A* 算法的全部路径搜索过程

因此,反过来从目标点“E”开始沿着指针的方向逐步地通过各个点的“父点”到达起始点“S”的路径就是得到的最优路径。最终的结果见图 3-7。

A* 算法总结如下。

(1) 将起始点加入到栅格开放列表中。

(2) 重复以下操作。

①选择在栅格开放列表中拥有最低 F 值的栅格作为当前的栅格。

②将所选中的栅格加入到栅格关闭列表中。

③对当前所选中的栅格的 8 个临近栅格做如下操作：

- 如果该栅格为非可通过栅格或者在栅格关闭列表中的话则忽略该栅格。否则做接下来的操作。
- 如果该栅格不在栅格开放列表当中，则将其加入到栅格开放列表当中。将当前所选中的栅格标记为该栅格的“父点”。记录该栅格的 F 值、 G 值和 H 值。
- 如果该栅格已经在栅格开放列表当中的话，则查看当前所得到的路径是否更好。这里，更低的 G 值意味着更好的路径。如果是这样的话，则将该栅格的父点变更为当前所选中的栅格，然后重新计算该栅格的 F 值和 G 值。

④当将目标点加入到栅格关闭列表中时，或栅格开放列表为空时停止搜索。

(3) 保存得到的最优路径。

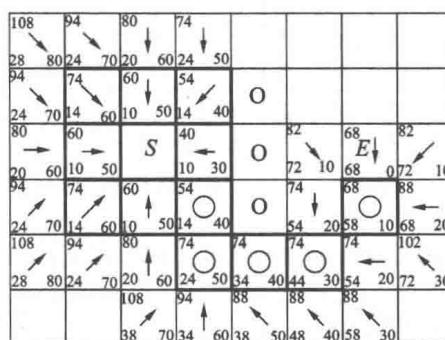


图 3-7 利用 A* 算法得到的最终路径

3.2 蚁群算法

1. 蚁群的行为

通过对自然界真实蚂蚁活动的观察，通常可以发现这一现象：蚂蚁总是习惯成群结队地活动。单只蚂蚁个体拥有的行为非常简单，然而，由这些简单个体所组成的群体，却能表现出非凡的能力，经常能完成一些看似蚂蚁不可能完成的任务。这一情况的产生，其根本原因是什么呢？主要就是由于不同种类的蚂蚁是具有明确的组织与分工的，它们相互之间通力协作，使蚂蚁简单的个体行为汇集成了整个蚂蚁王国的群体行为。

在昆虫世界中，蚂蚁的组成是一种群居的世袭大家庭，我们称之为蚁群。蚂蚁分为世袭制的蚁王（后）和工蚁两种，它们具有高度组织的社会性，彼此沟通不仅可以借助触觉和视觉的联系，在大规模的协调行动中还可以借助外激素（或称信息

素)之类的信息介质。

蚂蚁的觅食行为就是蚂蚁群体行为的主要表现之一。生物学家通过长期的观察和研究发现,虽然蚂蚁的食物源一般总是随机地散落在蚁巢周围,但是当蚁群在寻找食物时,它总是有能力在没有任何可见提示条件下找到一条从蚁巢到食物源的最短路径,而且当环境发生改变时,蚁群还能适应其变化,重新调整,搜索到新的最短路径。首先我们要理解蚂蚁是如何觅食的,蚂蚁平时在巢穴附近作无规则行走,一旦发现食物,并不立即进食,而是将之搬回蚁穴与其他蚂蚁分享,在食物小时则独自搬回蚁穴,否则就回蚁穴搬兵,一路上会留下外激素,食物越大外激素的浓度就越大,越能吸引其他的蚂蚁过去一起搬食物,这样最终就能将食物全部搬回蚁穴。这个过程用程序实现看似非常复杂,要编写一个“智能”的蚂蚁也看似不太可能,事实上每个蚂蚁只做了非常简单的工作:检查某个范围内有无食物,并逐渐向外激素浓的方向运动。简而言之,蚁群运动无非是同时反复执行多个简单规则而已。下面详细说明蚁群中的这些简单规则。

①范围:蚂蚁观察到的范围是一个 3×3 方格世界,并且能移动的距离也在这个范围之内。

②环境:蚂蚁所在的环境是一个虚拟的世界,其中有障碍物,有别的蚂蚁,还有外激素。外激素有两种,一种是找到食物的蚂蚁撒下的食物外激素,另一种是找到窝的蚂蚁撒下的窝的外激素。每个蚂蚁都仅仅能感知它范围内的环境信息。环境以一定的速率让外激素消失。

③觅食规则:在每只蚂蚁能感知的范围内寻找是否有食物,如果有就直接过去。否则看是否有外激素,并且在能感知的范围内比较哪一点的外激素最多,这样,它就朝外激素多的地方走,并且每只蚂蚁会以小概率犯错误,从而并不是往外激素最多的点移动。蚂蚁找窝的规则和上面一样,只不过它对窝的外激素做出反应,而对食物外激素没反应。

④移动规则:每只蚂蚁都朝向外激素最多的方向运动,并且,当周围没有外激素指引的时候,蚂蚁会按照自己原来运动的方向惯性地运动下去,并且,在运动的方向有一个随机的小的扰动。为了防止蚂蚁原地转圈,它会记住最近刚走过了哪些点,如果发现要走的下一点最近已经走过了,它就会尽量避开。

⑤避障规则:如果蚂蚁要移动的方向有障碍物挡住,它会随机地选择另一个方向,并且有外激素指引的话,它会按照觅食的规则行动。

⑥播撒外激素规则:每只蚂蚁在刚找到食物或者窝的时候播撒的外激素最多,并随着它走的距离越远,播撒的外激素越来越少。

根据这几条规则,蚂蚁之间并没有直接的关系,但是每只蚂蚁都和环境发生交互,而通过外激素这个纽带,实际上把各个蚂蚁关联起来了。比如,当一只蚂蚁找到了食物,它并没有直接告诉其他蚂蚁这儿有食物,而是向环境播撒外激素,当其

他的蚂蚁经过它附近的时候,就会感觉到外激素的存在,进而根据外激素的指引找到了食物。成功的觅食算法正是使搜索食物的时间最小化。

Goss S 等在 1989 年做了一个有名的“非对称双桥实验”^[84]。实验中,将蚁巢和食物源通过一个非对称的双支桥相连,然后把蚂蚁放置于蚁巢,并使其向食物源移动进行觅食,经过一段时间后观察两个分支路径中蚂蚁的分布情况。如图 3-8 所示,图(a)表示实验中的非对称双支桥及其尺寸,图(b)、图(c)分别表示实验进行到 4 min 和 8 min 后蚂蚁的分布情况。实验结果清晰表明,蚂蚁在经过一段时间的搜索之后,最终总是能找到一条从巢穴通往食物源的最短路径。

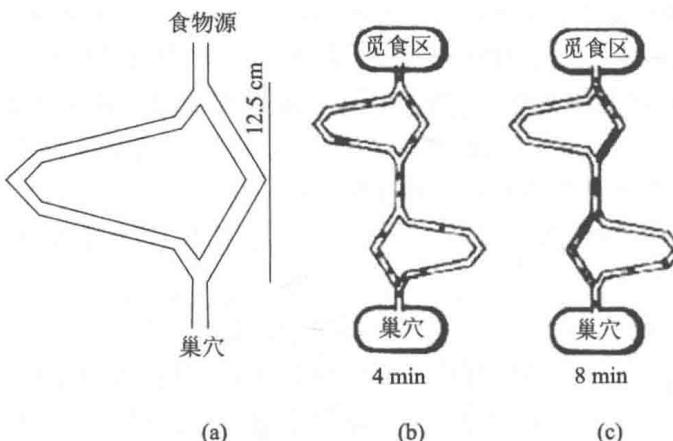


图 3-8 “非对称双桥”实验

2. 蚁群算法的基本思想

如前所述,蚁群觅食,能够找到最短路径的核心是蚂蚁对信息素的释放和感知。在觅食过程中,当蚂蚁到达某个路口,若此时待选择的各条路径上信息素强度相同,那么选择任意一条路径的概率也就相同,蚂蚁就会随机地进行路径转移;若待选择路径上的信息素强度存在差异,那么蚂蚁总是会倾向于选择信息素强度高的路径,运动的同时还会继续释放出信息素。当后续蚂蚁再次来到这个路口时,由于信息素强度较高的路径对蚁群的吸引更大,从而使选择该路径的蚂蚁越来越多,导致的结果是:该路径上的信息素累积越来越多,后续被选中的概率越来越大,最终使所有蚂蚁都会形成一定趋势沿着该路径运动。

为了便于理解蚂蚁觅食最短路径的形成过程,下面通过简化两条路径的情况,图示形象化地进行说明。

如图 3-9 所示,假设蚁巢和食物源之间有一障碍物,从蚁巢到达食物源的路径有两条:蚁巢—A—B—D—食物,蚁巢—A—C—D—食物。其中路径长度 $AB=BD=1$, $AC=CD=2$, 蚁巢到 A 点和食物到 D 点的距离也为 1。假定蚁群一共有 40 只蚂蚁共同觅食,单位时间内蚂蚁移动的距离设置为 1, 蚂蚁每走过单位路径,

留下的信息素也设置为 1, 初始时刻令所有路径上的信息素均为 0。

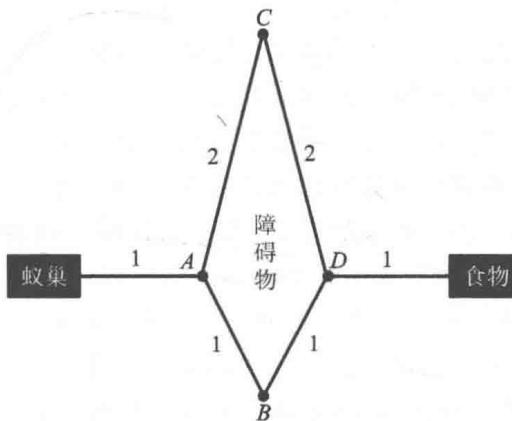


图 3-9 蚁群觅食行为示意图

初始时刻, 40 只蚂蚁从蚁巢出发。当它们到达 A 点时, 遇到障碍物, 此时到底是往 B 点走还是往 C 点走, 就面临了路径的选择。而由于初始时刻路径上的信息素均为零, 所以蚂蚁对于选择 AB 或 AC, 概率是相等的。因此可以从整体上认为, 20 只蚂蚁走 AB, 20 只蚂蚁走 AC; $t=4$ 时刻, 选择走 AB 路径的 20 只蚂蚁(设定为第一组蚂蚁)已经到达了食物源取得食物, 随即准备返回蚁巢; $t=5$ 时刻, 选择走 AC 路径的 20 只蚂蚁(设定为第二组蚂蚁)到达 D 点, 而此时第一组蚂蚁返回恰好到达 D 点, 两组蚂蚁相遇。此时 DB, DC 上的信息素强度相同, 均为 20, 因此第一组返回蚂蚁又以相同概率选择 DB 和 DC 路径, 平均来看 10 只走向 DB, 10 只走向 DC; $t=6$ 时刻, 第二组蚂蚁到达食物源并开始返回; $t=7$ 时刻, 第二组蚂蚁同样返回 D 点, 此时 DB, DC 上的信息素强度均为 30, 因此同样看 10 只选择 DB, 10 只选择 DC; $t=8$ 时刻, 第一组选择 ABD 路径来回的 10 只蚂蚁到达蚁巢, 此时 AC 中点、DC 中点以及 B 点各有 10 只蚂蚁; $t=9$ 时刻, 第一组最先到达蚁巢的 10 只蚂蚁再次出发到达 A 点, 再次面临 AB 和 AC 路径的选择。此时 AB 段的信息素为 40, AC 段为 30, 因为蚂蚁总是倾向于选择信息素强度高的路径, 所以 AB 段将会以较大概率被选中, 或者说会有较多的蚂蚁选择 AB 而放弃 AC。而此动作进一步增强了 AB 段的信息素强度。随着蚁群觅食过程进一步进行, ABD 段路径上的信息素强度会越来越大, 被蚂蚁选择前进的概率也越来越大, 最终将会使蚁群完全选择沿着最短路径“蚁巢—A—B—D—食物”来回觅食。

蚁群算法是一类仿生优化算法, 是通过模拟蚂蚁行为来解决实际问题的算法。在算法应用中的蚂蚁我们称为“人工蚂蚁”, 显然它是对真实蚂蚁的模拟, 它应该具备真实蚂蚁的基本特性: 比如个体之间相互交流, 协同完成共同任务, 能够根据当前积累的经验选择路径, 等等。其中, 还有一个相当重要的因素不容忽略: 真实蚁

群在觅食过程中所释放的信息素会随着时间的推移而挥发甚至消失,这是生物自然特性所决定的。而正是由于信息素的这种挥发特性,才使蚂蚁更能有效地找到最短路径。因此,人工蚂蚁也必须具备这种特性。当然,人工蚂蚁还具备了一些真实蚂蚁不具有的一些特性,比如人工蚂蚁存在于一个离散的空间,它们的移动是从一个状态到另一个状态的转换;人工蚂蚁释放信息素的时间和强度可以根据具体情况而定,但真实蚂蚁是在移动的同时释放信息素;人工蚂蚁的转移可以不是盲目的,还可以受到很多其他环境因素的启发;人工蚂蚁还可以具有路径预测、局部搜索等人为赋予的真实蚂蚁所不具备的功能。

蚁群的觅食行为还充分体现出一种分布式机制。将蚂蚁觅食看作一个共同的任务,那么每一个蚂蚁个体都在为这一任务而努力工作。单个蚂蚁在觅食过程中所做出的只是很小一部分贡献,它所找到的路径也极有可能不是最短的觅食路径。但正是因为有了若干蚂蚁部分贡献的积累,才最终导致蚁群表现出找到最短觅食路径的能力。蚁群算法作为对蚁群觅食行为的抽象,其中必然体现出一种分布式机制。在实际问题的求解过程中,算法中的人工蚂蚁各自独立地针对问题进行解空间的搜索,每只蚂蚁找到的候选解间接地影响着最优解的进化,但并不一定直接决定着最优解的最终形成。蚁群算法所表现出来的这种分布式机制对问题最优解的搜索是在空间的多个点同时进行,这样有效弥补了某些从单一点出发的搜索算法的缺陷,不仅可以提高算法的可靠性,而且往往能够寻找到问题的更满意的解。

3. 基本蚁群算法原理

蚁群算法(ant colony optimization, ACO),又称蚂蚁算法,是一种用来在图中寻找优化路径的概率型算法^[85]。它由 Marco Dorigo 于 1992 年在他的博士论文中引入,其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。蚁群算法是一种模拟进化算法。初步的研究表明该算法具有许多优良的性质。针对 PID 控制器参数优化设计问题,将蚁群算法设计的结果与遗传算法设计的结果进行了比较,数值仿真结果表明,蚁群算法具有一种新的模拟进化优化方法的有效性和应用价值。蚁群算法是一种求解组合最优化问题的新型通用启发式方法,该方法具有正反馈、分布式计算和富于建设性的贪婪启发式搜索的特点^[86]。蚁群算法模拟了蚂蚁群体觅食行为,是一种新的计算智能模式,它基于如下基本假设。

①蚂蚁之间通过信息素和环境进行交互。每只蚂蚁仅根据其周围的局部环境作出反应,也只对其周围的局部环境产生影响。

②蚂蚁对环境的反应由其内部模式决定。因为蚂蚁是基因生物,蚂蚁的行为实际上是其基因的适应性表现,即蚂蚁是反应性主体。

③在个体水平上,每只蚂蚁仅根据环境做出独立选择。在群体水平上,单只蚂蚁的行为是随机的,但蚂蚁可通过自组织过程,形成高度有序的群体行为。

由上述假设和分析可见,基本蚁群算法的寻优机制包括两个基本阶段:适应阶段和协作阶段。在适应阶段,各候选解根据积累的信息不断调整自身的结构,路径上经过的蚂蚁越多,信息素越多,该路径越容易被选择;时间越长,信息素会越少。在协作阶段,候选解之间通过信息交流,以期望产生性能更好的解,类似于自动学习机的学习机制。

蚁群算法实际上是一种多智能体系统,其自组织机制使得蚁群算法不需要对所求问题的每一方面都有详尽的认识^[87]。蚁群算法的自组织机制本质上是在没有外界作用下使系统熵增加的动态过程,体现了从无序到有序的动态演化,其逻辑结构如图 3-10 所示。

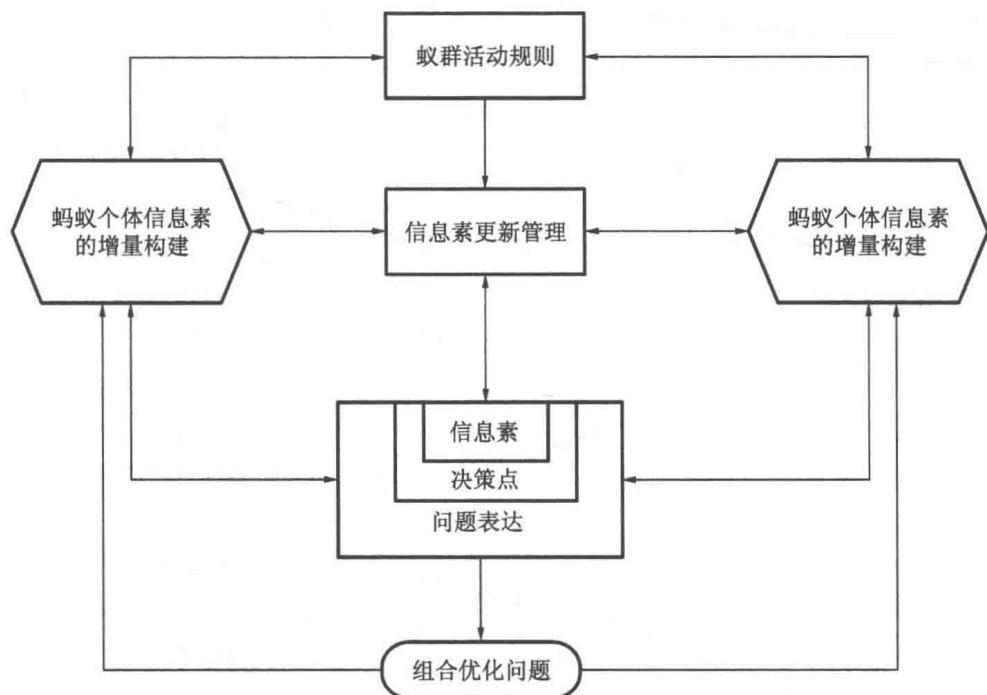


图 3-10 基本蚁群算法的逻辑结构

由图 3-10 可见,先将具体的组合优化问题表述成规范的格式,然后利用蚁群算法在“探索”和“利用”之间根据信息素这一反馈载体确定决策点,同时按照相应的信息素更新规则对每只蚂蚁个体的信息素进行增量构建,随后从整体角度规划出蚁群活动的行为方向,周而复始,即可求出组合优化问题的最优解。

4. 基本蚁群系统模型

1) TSP 问题(traveling salesman problem)描述

蚁群算法最早是基于 TSP 问题提出并成功应用的^[88],下面首先简单介绍一下 TSP 问题。

TSP 问题属于数学领域中的经典问题。它的历史十分悠久,最早的描述出现在著名数学家欧拉研究的骑士周游问题上,即走完国际象棋棋盘中所有的 64 个方格一次且仅有一次的周游路径。TSP 问题一般可以描述为:假设有 n 个城市,有一个旅行商需要从某一个城市出发,拜访完所有城市后再回到原出发城市,其中要求每个城市只能拜访一次,要求找到一条最短的巡回路径。

结合图论知识,可将 TSP 问题用数学模型表示:

设 $V = \{v_1, v_2, v_3, \dots, v_n\}$ 表示 n 个城市的集合, $E = \{E_{ij} \mid v_i, v_j \in V\}$ 是集合 V 中元素(城市)两两连接的集合, d_{ij} ($i, j = 1, 2, \dots, n$) 表示 E_{ij} 的长度, 即

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3-1)$$

$G = (V, E)$ 是一个有向图, TSP 中找出长度最短的哈密顿 (Hamilton) 回路, 即一条对 $V = \{v_1, v_2, v_3, \dots, v_n\}$ 中 n 个城市访问一次的最短封闭曲线。

TSP 是一类 NP 难问题,对于其最短路径的求解,目前已有很多种智能算法能够实现。按照城市之间往返距离的异同, TSP 问题可分为对称 TSP 和非对称 TSP 两大类。若往返距离相同,即 $d_{ij} = d_{ji}$, 则为对称 TSP; 若 $d_{ij} \neq d_{ji}$, 则为非对称 TSP。为了方便建立模型,本文所涉及的均为对称 TSP 问题。

2) 基本蚁群算法数学模型

为了建立数学模型,首先假设蚁群中的蚂蚁具有以下特征。

每只蚂蚁能够在给定 TSP 问题的城市之间进行转移,并且根据以包含城市之间距离和信息素强度为变量的概率函数选择下一个转移城市。规定蚂蚁在一次循环内所有城市只能访问一次。蚂蚁在进行城市转移的同时能够释放相应强度的信息素。

设 n 表示 TSP 问题的规模,即城市数量, m 表示蚁群中的蚂蚁数量, d_{ij} ($i, j = 1, 2, \dots, n$) 表示城市 i 与城市 j 之间的距离。初始时刻,各条路径上的信息素强度相等,蚂蚁 k ($k = 1, 2, \dots, m$) 在前进过程中,根据各条路径上的信息素强度决定其转移方向。为了使蚂蚁在一次循环过程中同一城市只能被访问一次,需要建立一个数据结构(禁忌表)来记录当前已经访问过的城市,而没有访问到的城市则暂时不写入禁忌表。通常禁忌表用 tabu_k ($k = 1, 2, \dots, n$) 来表示。当一次循环访问结束之后,需要将禁忌表清零,以便蚂蚁能够进入下一次迭代搜索。

那么,在搜索过程中,某一时刻 t 蚂蚁 k 由城市 i 转移到城市 j 的状态转移概率 p 可表示为

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ik}(t)]^\alpha [\eta_{ik}]^\beta}, & \text{如果 } j \in \text{allowed}_k \\ 0, & \text{否则} \end{cases} \quad (3-2)$$

式中: $\text{allowed}_k = \{V - \text{tabu}_k\}$ 表示蚂蚁 k 下一步允许选择转移的城市; $\tau_{ij}(t)$ 表示

在 t 时刻城市 i 和城市 j 之间的路径上的信息素强度; η_{ij} 表示路径边 (i, j) 的能见度, 用启发函数计算, TSP 问题中的表达式为 $\eta_{ij}(t) = \frac{1}{d_{ij}}$; α 为信息启发式因子, 表示了蚂蚁在运动过程中所积累的信息素量在指导蚁群搜索中的相对重要程度, 其值越大, 则蚂蚁越倾向于选择信息素强度高的路径进行转移; β 为期望启发式因子, 表示了启发信息在蚂蚁搜索路径过程中的受重视程度, 其值越大, 则蚂蚁越倾向于选择当前较短的路径的城市进行转移。

在蚂蚁搜索前进过程中, 一方面每只蚂蚁会在各条路径上留下相应的信息素, 另一方面各路径边上的信息素会随着时间的推移逐渐挥发。因此, 需要对信息素进行更新处理。

在每只蚂蚁走完一步或者完成一次周游(访问所有 n 个城市)后, 可以对残留信息素进行更新处理。 $t+1$ 和 $t+n$ 时刻在路径 (i, j) 上的信息素量更新规则公式如下。

$$\begin{cases} \tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \\ \Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \end{cases} \quad (3-3)$$

$$\begin{cases} \tau_{ij}(t+n) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \end{cases} \quad (3-4)$$

式中, ρ 表示信息素挥发系数, 则 $1 - \rho$ 表示信息素残留系数, $\rho \in [0, 1]$; $\Delta\tau_{ij}(t, t+1)$ 表示时刻 $(t, t+1)$ 路径 (i, j) 上的信息素总的增量, $\Delta\tau_{ij}^k(t, t+1)$ 表示第 k 只蚂蚁在时刻 $(t, t+1)$ 释放在路径 (i, j) 上的信息素量; $\Delta\tau_{ij}$ 表示本次循环中路径 (i, j) 上的信息素增量, $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在本次循环中释放在路径 (i, j) 上的信息素量。

根据信息素更新方式的不同, M. Dorigo 提出了三种不同的基本蚁群算法模型, 分别称为蚁量(Ant-quantity)模型、蚁密(Ant-density)模型和蚁周(Ant-cycle)模型, 三种模型的差别仅在于 $\Delta\tau_{ij}^k$ 的表达式不同^[89]。

Ant-quantity 模型:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}}, & \text{如果蚂蚁 } k \text{ 在时间段 } t \text{ 到 } t+1 \text{ 中从 } i \text{ 点走到 } j \text{ 点} \\ 0, & \text{否则} \end{cases} \quad (3-5)$$

Ant-density 模型:

$$\Delta\tau_{ij}^k = \begin{cases} Q, & \text{如果蚂蚁 } k \text{ 在时间段 } t \text{ 到 } t+1 \text{ 中从 } i \text{ 点走到 } j \text{ 点} \\ 0, & \text{否则} \end{cases} \quad (3-6)$$

Ant-cycle 模型:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{如果蚂蚁 } k \text{ 走过的路径中包括 } i \text{ 点走到 } j \text{ 点的连线} \\ 0, & \text{否则} \end{cases} \quad (3-7)$$

在以上三个模型中, Q 均表示一常数, 它在一定程度上影响算法的收敛性能。对于 Ant-quantity 和 Ant-density 模型, 信息素都是在蚂蚁完成一步转移后就进行更新操作的, 采用公式(3-3)实现信息素更新; 而在 Ant-cycle 模型中, 信息素是在蚂蚁完成一次循环后才进行更新操作的, 循环中路径 (i, j) 上的信息素增量为 $\frac{Q}{L_k}$, 其中 L_k 表示第 k 只蚂蚁在本次循环中所走路径的总长度, 采用公式(3-4)来实现。经过一系列实验结果表明, Ant-cycle 模型的算法性能要优越于 Ant-quantity 和 Ant-density 模型。因此, 在解决实际问题时通常采用 Ant-cycle 模型。

3) 基本蚁群算法复杂度分析

根据算法复杂性理论, 评价一个算法质量的优劣, 主要从算法的时间复杂度和空间复杂度来考虑。算法的时间复杂度是指执行该算法的时间代价, 在实际应用中通常把算法的基本语句执行次数定义为算法的时间复杂度。如果一个问题的规模为 n , 通常用 $T(n)$ 表示某一算法解决这一问题需要的时间, $T(n)$ 则称为这一算法的时间复杂度。与时间复杂度类似, 一个算法的空间复杂度定义为该算法所耗费的存储空间, 主要是指执行该算法时计算机所需占用存储空间的度量, 它也是问题规模 n 的函数, 通常记作 $S(n)$ 。对于一个算法来说, 其时间复杂度和空间复杂度往往是相互影响的。当设计一个算法时, 需要综合考虑算法的各项性能(包括算法描述语言的特点、算法处理问题的规模、算法处理问题的时间要求、算法处理问题的结果倾向性要求、算法运行的系统环境等各方面因素)才能设计出比较优秀的算法。

基本蚁群算法的复杂度研究主要是指蚁群算法在求解实际问题时所需要的各種资源, 主要包括时间资源和空间资源^[90]。若 n 表示 TSP 问题的规模(城市数量), m 表示算法的蚂蚁数目, n_c 表示算法迭代次数, 那么基本蚁群算法的时间复杂度表示为 $T(n)=O(n_c \cdot n^2 \cdot m)$, 此处“0”表示数量级的概念, 空间复杂度表示为 $S(n)=O(n^2)+O(n \cdot m)$ 。

4) 蚁群算法的特征

系统学是一门较新的学科, 其基本特点是强调整体性。系统学十分庞大, 由于不同学科的研究范围和侧重点不同, 往往对“系统”给出了不同的定义。在基础科学层次, 通常采用系统学创始人 Beralanffy L V 所给出的定义: 系统可以确定为处于一定的相互关系中并与环境发生关系的各组成部分(要素)的综合体。这个定义强调的不是功能, 而是系统元素之间的相互作用以及系统对元素的整体作用。该定义可更为精确地表述为, 如果对象 S 满足以下两个条件:

- ① S 中至少包含两个不同的对象;
- ② S 中的对象按照一定的方式互相联系在一起。

则称 S 为一个系统, 同时定义 S 中的个体对象为系统的元素。

很显然,自然界中的蚁群具备了系统的三个基本特征,即多元性、相关性和整体性,从而构成了一个系统^[91]。在这个系统中,蚂蚁的个体行为是系统的元素,蚁群行为的相互影响体现了系统的相关性,而蚁群可以完成个体完成不了的任务则体现了系统的完整性,显示出系统整体大于部分之和的整体突现原理。

作为对蚁群觅食行为抽象的蚁群算法,如果把算法本身看作一个整体,就会发现它具备了系统的特征,这也是人工智能算法最重要的特征之一。所以蚁群算法是一个系统。它具有如下特征^[91]:

(1) 分布式计算。

生命系统是一个分布式系统,它使得生命体具有更强的适应能力;蚁群算法作为对蚁群觅食行为的抽象,体现了群体行为的分布式特征。每只蚂蚁在问题空间的多个点同时开始相互独立地构造问题解,而整个问题的求解不会因为某只蚂蚁无法成功获得解而受到影响,不仅使得算法具有较强的全局搜索能力,也增加了算法的可靠性。

(2) 自组织。

蚁群算法的另一个重要特征是自组织,这也是其他人工智能算法(如遗传算法、人工神经网络、微粒子群、人工免疫算法、人工鱼群算法等)共有的基本特征。

在系统论中,组织行为可分为自组织和他组织两大类,其根本区别在于组织力或组织指令是来自系统内部还是来自系统外部,前者称为自组织,而后者称为他组织。从抽象意义上讲,自组织就是在没有外界作用下使得系统熵增加的过程(即系统从无序到有序的进化过程)。如蚂蚁、蜜蜂这样的昆虫,由于个体作用简单,而个体之间的协作作用特别明显,因而可把它们当作一个整体来研究,甚至把它们看作一个独立的生物体。在这样的群落中,生物个体相互作用,协同完成某项群体工作,自然体现出很强的自组织特性,蚁群算法就体现了这一过程。

自组织大大增加了算法的鲁棒性,传统的算法都是针对某一具体问题而设计的,这往往建立在对该问题有了全面清晰认识的基础上,通常很难适应其他问题。而自组织的蚁群算法不需要对待求解问题的所有方面都有所认识,因此较容易应用到一类问题中。

(3) 正反馈。

在系统学上认为,反馈就是把系统现在的行为作为影响系统未来行为的原因,它代表信息输入输出的反作用。反馈分为正反馈和负反馈两种。

对于自然界中的蚂蚁而言,蚂蚁能够最终找到最短路径,直接依赖于最短路径上的信息素的堆积,而信息素的堆积却是一个正反馈的过程。对于蚁群算法而言,初始时刻在环境中存在完全相同的信息素,给系统一个微小的扰动,使得各个边上信息素量大小不同,蚂蚁构造的解就存在了优劣。算法采用的反馈方式在较优解经过的路径留下更多的信息素,更多的信息素又吸引了更多的蚂蚁,这个正反馈的

过程使得初始值不断地扩大,同时又引导整个系统向着最优解的方向进行。

蚁群算法中同样隐含着负反馈机制,它体现于蚁群算法在构造问题解的过程中用到了概率搜索技术,通过该技术增加了生成解的随机性。随机性的影响就在于接受了解在一定程度上的退化,另一方面又使得搜索范围得以在一段时间内保持足够大。这样正反馈缩小搜索范围,保证算法朝着最优解的方向进化;而负反馈保持搜索范围,避免算法过早收敛于不好的结果。恰恰是在正反馈和负反馈共同作用的影响下,蚁群算法才得以自组织地进行,从而得到问题在一定程度上的满意解。值得一提的是,在已经公开发表的大部分论文中,对于蚁群算法所隐含的负反馈机理没有提及,但这并不影响对蚁群算法本质特征的理解以及对蚁群算法的改进和广泛应用。

5) 蚁群算法的优点与不足

(1) 蚁群算法的优点。

蚁群算法是一种结合了分布式计算、正反馈机制和贪婪式搜索的算法,具有很强的搜索较优解的能力。正反馈能够快速地发现较优解,分布式计算避免了早熟收敛,而贪婪式搜索有助于在搜索过程的早期阶段找出可接受的解决方案,缩短了搜索时间。尽管蚂蚁个体比较简单,但整个蚂蚁群体却表现为高度机构化的社会组织,在许多情况下能完成远远超过蚂蚁个体能力的复杂任务,这种能力来源于蚂蚁群体中的个体协作行为,因此具有如下优点。

①较强的鲁棒性:对蚁群算法模型稍加修改,可以应用于其他问题。

②分布式计算:蚁群算法是一种基于种群的人工智能算法,其本质具有并行性,易于并行实现。

③易于与其他方法结合:蚁群算法很容易与其他算法结合,以改善算法的性能。

(2) 蚁群算法的不足之处。

当然,这种算法也存在一些不足之处,Marco Dorigo 曾指出以下几点不足之处^[92]。

①蚁群算法需要较长的搜索时间。

虽然蚂蚁通过信息交换能够找到比较优的解,但是当问题规模较大时,不可避免地会增加搜索所需时间。

②容易陷入局部最优解即停滞现象。

当蚁群搜索进行到一定程度后,所有个体所发现的解完全一致,不能再对解空间进行进一步搜索,不利于发现更好的解。

③蚁群算法是一种概率算法。

从数学上证明它的正确性与可靠性还是比较困难的,而且蚁群算法在解决问题的时候,算法系统的高层次的行为是需要通过低层次的蚂蚁之间的简单行为交

互涌现产生的。单个蚂蚁控制简单并不意味着整个系统设计简单,设计者必须能够将高层次的复杂行为(也就是系统所需要执行的功能,例如,旅行商问题、车辆路径问题、图着色问题等)映射到低层次的蚂蚁的简单行为(例如,信息素的释放)上面,而这二者之间是存在较大差别的。并且在系统设计时也要保证多个个体简单行为的交互能够涌现出我们所希望看到的高层次的复杂行为。这可以说是蚁群算法乃至群集智能中一个极为困难的问题。

6) 路径规划蚁群算法

通常情况下,用于路径规划的蚁群算法多以栅格法作为基础,将地图按照一定的分辨率划分成一系列栅格,航行空间就被限定在栅格所定义的点上。这样处理之后,整个地图的位置离散信息相当于一组二维矩阵,对于后期计算处理非常直观而且方便。因而,栅格法也成为了一种最常使用的方法。

在使用栅格法的算法中,比较典型地分为两类:旋转坐标栅格模型和原地图坐标栅格模型。前者的优势在于可利用旋转后的X轴正方向作为目标指向,使得算法运算到最后总能使蚂蚁运行到终点的位置上。但是这种算法的缺点是仿真中的每一单步计算量大,栅格分辨率一旦提高将带来巨大的计算量负担。原地图坐标栅格模型中,每一单步只使用当前栅格周围的栅格信息来计算(一般分为4邻域法和8邻域法),因此单步运算量比旋转坐标栅格模型小很多。然而这样一来,这种模型在计算的时候可搜索空间大大增加,也造成了求解过程中的随机性增加,因此算法整体收敛速度会受影响。两种模型各有利弊,可根据具体应用的不同需要,合理选择合适的模型。

这里引入以下描述符号。

定义:W为蚁群中蚂蚁的个数; $d_{i,j}$,($i=1,2,\dots,n_1$; $j=1,2,\dots,n_2$; n_1 和 n_2 分别是对平面工作环境的2维划分维数)为平面环境中位置点i与j之间的距离; $b_i(t)$ 为t时刻位于位置点i处的蚂蚁数目; $\tau_{i,j}(t)$ 表示t时刻在路径(i,j)上残留的信息素的量。则有等式:

$$W = \sum_{i=1}^n b_i(t) \quad (3-8)$$

因为在初始时刻,每条路径的信息素的量都是相等的,所以有预设条件 $\tau_{i,j}(t)=C$ (C是一常数)。

蚂蚁 k ($k=1,2,\dots,W$)在运动过程中,会根据各条路径上的信息素量决定其下一步的转移方向。在时刻t时,蚂蚁 k 要从位置点i向j转移,其对应的转移概率可定义为

$$p_{i,j}^k(t) = \begin{cases} \left[\tau_{i,j}^a(t) \eta_{i,j}^b(t) \right] / \left[\sum_{\gamma \in S_i^k} \left[\tau_{i,\gamma}^a(t) \eta_{i,\gamma}^b(t) \right] \right], & j \in S_i^k \\ 0, & \text{否则} \end{cases} \quad (3-9)$$

式中: $\eta_{i,j}(t)$ 是局部能见度启发函数(在这个问题中定义为 $\frac{1}{d_{i,j}}$); 参数 α 和 β 分别表示 $\tau_{i,j}(t)$ 和 $\eta_{i,j}(t)$ 对整个转移概率的影响权值; S_i^k 表示蚂蚁 k 在位置点 i 处的可行邻域(即与点 i 相邻且尚未被蚂蚁 k 访问过的位置点的集合), 借助于种群的记忆功能, 这个集合在进化过程中将会不断地动态调整。

随着时间的推移, 信息素将会逐渐地挥发。这里, 用 ρ 表示在某条路径上信息素挥发后的剩余度。在经过 h 时刻后, 蚁群会完成一个循环的移动。此时, 各条路径上信息素的量将按照全局调整准则进行调整。

$$\tau_{i,j}(t+h) = \rho \tau_{i,j}(t) + \sum_{k=1}^W \Delta \tau_{i,j}^k \quad (3-10)$$

式中: $\Delta \tau_{i,j}^k$ 表示蚂蚁 k 在本次循环中留在路径 (i, j) 上的信息素的量, 可基于局部调整准则对其定义为

$$\Delta \tau_{i,j}^k = \begin{cases} \frac{Q}{L_k}, & \text{如果蚂蚁 } k \text{ 在本次循环中通过路径 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (3-11)$$

式中: Q 为表示信息素强度的一定常数; L_k 表示蚂蚁 k 在本次循环中经过所有路径的长度。

在初始时刻, 有 $\tau_{i,j}(0) = C$, $\Delta \tau_{i,j}^k = 0$ ($i = 1, 2, \dots, n_1$; $j = 1, 2, \dots, n_2$; $k = 1, 2, \dots, W$)。此外 $\tau_{i,j}$, $\Delta \tau_{i,j}^k$, $p_{i,j}^k(t)$ 的表达式也可根据算法的具体应用而有所调整。

算法的步骤如下。

步骤 1: 产生初始时刻的蚂蚁种群移动路径。根据移动过程中途经各点周围的距离启发式信息概率, 产生多条从起点到终点的可行移动路径, 每一条路径代表了一只蚂蚁的爬行轨迹。

步骤 2: 信息素的调整。对所产生的每一条可行移动路径, 分别计算路径的长度和所对应信息素的增量, 再采用设计的信息素轨迹更新函数对路径上各点所对应的信息素进行更新。

步骤 3: 对产生的每一可行路径进行一定的修正处理。例如, 可以将蚂蚁所走的弯曲路径逐段拉直为一条由直线段连接的可行路径(即成为一折线)。将此可行路径与记录的目前最短路径进行比较, 如果路径长度更小, 则用该路径替换最短路径。对路径上的所有点的信息素也根据步骤 2 中的方法进行更新。如果当前时刻已达到预先设定的终止时刻, 则转步骤 5。

步骤 4: 产生下一时刻的蚂蚁路径。综合使用当前点周围的距离启发式信息概率和基于信息素轨迹的转移概率, 产生由起点到终点的可行路径, 并转步骤 2。

步骤 5: 算法结束。将当前路径作为最短路径输出。

需要说明的是:在步骤4中提到了称为“距离启发式”的一个概率函数。这是一个类型如同下式的函数:

$$p_{i,j}^k = \begin{cases} \frac{\{[D_{\max}(i), e - D(j), e]W + U\}^Y}{\sum\limits_{j \in Dp(i)} \{[D_{\max}(i), e - D(j), e]W + U\}^Y}, & j \in Dp(i) \\ 0, & \text{否则} \end{cases} \quad (3-12)$$

这一函数只应用在原地图直角坐标栅格模型中。其作用在算法的初始阶段,当环境中还没有足够的信息素,或者在算法运行中,蚂蚁行进到没有信息素标识的区域,作为终点位置的“指向标”,引导蚂蚁总是以较大概率向能靠近终点的方向上移动。而在旋转坐标系的模型中,其旋转后的X轴正方向本身就起到了这个“指向”作用,因此不需要这个概率函数。

式(3-12)决定了只依赖于距离信息时,编号为k的蚂蚁从当前点i向其周围一点的移动概率。Dp(i)是点i周围1个单位距离内非障碍区中点的集合,算法中蚂蚁只能向前、后、左、右4个方向移动,因此 $0 < Dp(i) \leq 4$ 。D(j),e是从j点到终点e的距离,其值在算法执行前被预先计算出。

$D_{\max}(i), e$ 是 $D(j), e$ 中的最大值。因为 $Dp(i)$ 的各个 $D(j), e$ 之间相差不到2,所以需要对 $D(j), e$ 重新定标以体现它们之间的差别,否则启发式概率将变成一个随机函数,达不到应有的启发效果。这里引入3个标定系数W、U和Y用来增强距离启发式的作用。

另一个需要说明的是,参数 α 和 β 分别决定了信息素和启发式函数(能见度)的相对重要性。在一般的蚂蚁算法中, α 和 β 是常数,在算法执行过程中不作改变。但在路径规划问题中,由于蚂蚁可经过的点太多,很难确保每个点都获得信息素。这样将带来一个严重的问题,即如果获得信息素的点越少,那么结果陷入局部解的可能性就越大。事实证明,将 α 和 β 设为常数后,通常情况下不能找到较好的解。因此在路径规划一类应用中,通常都会将 α 和 β 设置为时间的函数。最普通的方法,是用一次函数来表达。由于普遍情况是在算法初始阶段,信息素的作用弱或者不起作用,而算法后期,算法对信息素的依赖又需要逐步增加。因此,一般会令 α 值随时间线性递增而 β 值随时间线性递减。但有时也会出现相反的设置或者设置 α 和 β 同时递增的情况,这可以根据算法的需要,改变调整。

最后要说明的是,在步骤4中提到的“综合使用当前点周围的距离启发式信息概率和基于信息素轨迹的转移概率”所采用的综合决定方法,被称为“比例选择策略”,也称作“轮盘赌策略”。通过比例选择,可交替使用三种概率“信息素的转移概率”、“距离启发式概率”和综合考虑上述两种概率所产生的“转移概率”,以决定下一个具体的可行移动点。

7) 轮盘选择

如图 3-11 所示,假设蚂蚁在 D 点, A, B, C 为已经去过的点, E, F, G 为没去过的点,现在蚂蚁要在 E, F, G 中选择一个点作为下一步要到达的点。

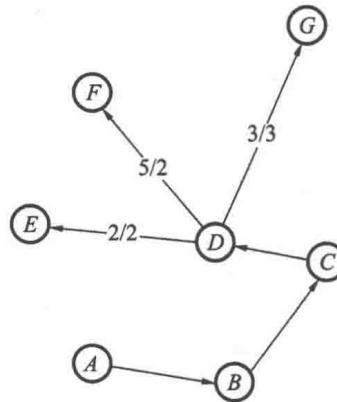


图 3-11 轮盘选择说明 1

假设: $\alpha=1.0, \beta=2.0, DE$ 间信息素为 2, DE 间距离为 2, DF 间信息素为 5, DF 间距离为 2, DG 间信息素为 3, DG 间距离为 3。那么 E, F, G 点被选择的概率分别是

$$p_E = \frac{2}{2 \times 2} = 0.5 \quad (3-13)$$

$$p_F = \frac{5}{2 \times 2} = 1.25 \quad (3-14)$$

$$p_G = \frac{3}{3 \times 3} = 0.3333 \quad (3-15)$$

总的概率为

$$p_{\text{total}} = p_E + p_F + p_G = 0.5 + 1.25 + 0.333 = 2.083 \quad (3-16)$$

那么 E, F, G 被选择的概率分别是

$$p^E = \frac{p_E}{p_{\text{total}}} = 24\% \quad (3-17)$$

$$p^F = \frac{p_F}{p_{\text{total}}} = 60\% \quad (3-18)$$

$$p^G = \frac{p_G}{p_{\text{total}}} = 16\% \quad (3-19)$$

如图 3-12 所示,如果蚂蚁选择概率最大的地方走,那么应该选择 F 点为下一个点。

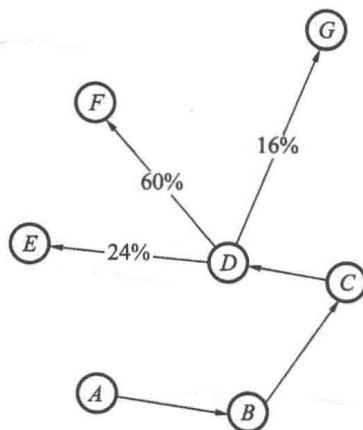


图 3-12 轮盘选择说明 2

这样选择会导致一个问题，每只蚂蚁到了 D 点都会选择 F 点作为下一个点，同样在其他的点上也会有这种情况发生，就是每只蚂蚁选择的下一个点都是同一个点，这会导致所有蚂蚁找到的路径都是相同的，会使蚂蚁失去探索新路径的机会，算法陷入停滞。

为了避免这个问题，就要使用轮盘选择来决定下一个点，方法如下：在 $[0,1]$ 之间取一个随机数 R 。然后用 R 减去 p^E ，如果减去后的结果小于等于 0 就选 E 作为下一个点，如果减去后还大于 0，就继续再减去 p^F ……直到减去后的结果小于等于 0 为止。然后用最后减去的那个概率对应的点作为下一个点。

如图 3-13 所示，假设 $[0,1]$ 之间取得的随机数字是 0.9，则深色的小箭头就顺时针旋转 0.9 圈。

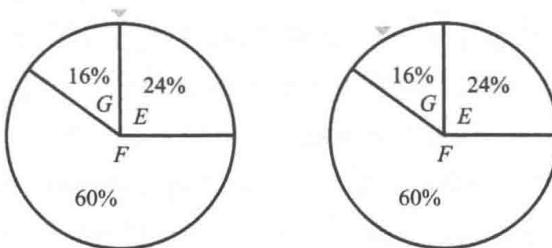


图 3-13 轮盘选择说明 3

可以看到深色的小箭头落在了 G 所在的扇区里，因此蚂蚁就选择 G 作为下一个点。从这个图中可以看出，在 $[0,1]$ 之间取一个随机数，这个数字落在 F 所在区

间里的可能性最大, E 次之, G 最小。可以看出使用轮盘选择可以使蚂蚁往概率大的地方走的可能性大, 但也有一定的可能往概率小的地方走, 这样可以使蚂蚁探索新的路径, 避免算法停滞或者进入局部最优解。

蚁群算法是带有正反馈的随机搜索算法, 随机性和正反馈是算法的核心, 丢失任何一个都会使算法失去作用。

正反馈是通过信息素来实现的, 而随机性正是通过轮盘选择来实现的, 不用轮盘选择, 算法就是贪婪搜索算法了。

5. 改进的路径规划蚁群算法

蚁群算法有如下的一些特性。①蚁群算法有分布式计算的特性。它有很强的全局搜索能力, 能够在大多数情况下提供全局最优解。并且, 分布式计算的特性使得蚁群算法在搜索过程中鲁棒性更强。②蚁群算法有自组织能力。这个特性可以增强蚁群算法的鲁棒性。③蚁群算法具有正反馈的特性。这可以提高系统的性能和可靠性, 从而提高蚁群算法的搜索效率。

蚁群算法在很多优化问题中有着广泛的应用。但是其也有自身的缺点, 包括停滞问题、容易陷入局部最优解问题、收敛速度慢问题等。当人工蚂蚁进入了一个凹形空间而无法再进行下一步搜索时便产生了停滞问题。解决这一问题的一个方法是采用回退原理^[93]。也就是说当人工蚂蚁下一时刻发生停滞现象时, 退回到前一个时刻所在的位置, 然后选择其他的路径继续搜索。文献[94]中作者针对人工蚂蚁的停滞现象采用了后退方式, 并且在后退过程中使得蚂蚁后退两步, 并把后退一步的栅格设为不选栅格, 这样避免了人工蚂蚁以后再次进入该栅格而形成死循环的情况。在文献[95]中作者引入了蚂蚁死亡的概念, 即当有蚂蚁出现停滞现象时, 该蚂蚁死亡。另外一种解决停滞现象的方法是凸化障碍栅格^[96]。对出现凹形空间的栅格区域进行凸化处理, 使得处理后的栅格地图中没有凹形空间, 从而避免了停滞现象的发生。文献[97]中作者对栅格地图进行了两次凸化处理, 从而使得人工蚂蚁避免陷入陷阱。在该方法中赋予了蚂蚁一定的视觉。在蚂蚁即将进入陷阱的前一个栅格时, 能够看见前面潜在产生停滞现象的栅格, 从而选择其他的可行栅格。容易陷入局部最优解的问题来自于蚁群算法的正反馈特性。当大量的人工蚂蚁集中在某一局部最优路径时, 该路径上的信息素会变得非常大, 从而会使得更多的人工蚂蚁选择该路径。在这种情况下能够搜索其他路径的人工蚂蚁会越来越少, 从而使得算法收敛到该局部最优解中。收敛速度慢和局部最优解问题是蚁群算法需要解决的两个主要问题。并且这两个问题是互相影响的。通常情况下, 为了避免陷入局部最优解需要扩大搜索的范围, 这样也就减慢了算法的收敛速度。

反之,为了加快算法收敛的速度,则必须减小搜索的范围。文献[94]中作者提出了根据路径间的偏差值来确定信息素的挥发系数的策略。在搜索初期,当各条路径的差异较大时,选取较大的挥发系数,从而使得人工蚂蚁搜索路径的多样性增加,改善了陷入局部最优解的问题;在算法的搜索后期,减小挥发系数,使得人工蚂蚁更容易选择搜索得到的最优路径,从而使算法更快地收敛。在文献[95]中作者采用了双蚂蚁思想。在起点和终点同时有人工蚂蚁进行搜索,从而增加了蚂蚁搜索的多样性,使得局部最优解问题得到改善。接下来介绍一种用得较多的蚁群改进方法^[98]——最大最小蚂蚁算法。在每次迭代搜索结束后仅对最优路径进行信息素更新,使得最优路径上的信息素会累积得比其他次优路径上的信息素多很多,从而加快了收敛速度。最大最小蚂蚁算法还限制了各路径上的信息素的最大值和最小值,以防止算法过快收敛。文献[96]中作者提出了一种结合遗传算法和蚁群算法路径搜索算法。在搜索初期先采用遗传算法得到初始阶段的信息素分布,然后利用蚁群算法进行最优路径的搜索。很多情况下利用蚁群算法得到的最优路径有很多的转折。这种情况下得到的最短路径往往并不是最优的路径。对于高速载重的移动机器人来说,路径上转折过多会使耗能过多,效率下降。多数情况下移动机器人需要在路径短的同时,希望转折能够比较少。文献[93]中作者提出了将机器人转角加入到最优路径的评价标准中,来得到比较平滑的路径的办法。文献[94]中作者采用惯性原理对所得到的最优路径进行再改造来得到更加平滑的路径。

在本书中提出了一种改进的蚁群算法,用来进行移动机器人的路径规划。改进的目的集中在前面提到的蚁群算法存在的四个主要问题上。在该改进算法中当人工蚂蚁陷入陷阱而导致停滞时,该人工蚂蚁会退回至前一个栅格上,并且将该“陷阱”点标记为“不要走”的点,从而使得其他人工蚂蚁不再选择该点,避免停滞现象再次发生。针对其他三个问题的改进方法详见下述的改进算法内容。

①设置算法中人工蚂蚁的数量为 m 。设 P_{start} 是出发点, P_{end} 是终点。算法开始时所有的人工蚂蚁都在出发点 P_{start} 上。将 P_{start} 放入禁忌集合 tabu_k ($k = 1, 2, \dots, m$) 中。其中 tabu_k 是人工蚂蚁 k 已经路过的点的集合。该集合在路径搜索过程中是动态调整的。初始状态的点 i 到点 j 之间路径上的信息素的值设为 $\tau_{ij}(0) = \tau_0$ (τ_0 是一常数)。设置系统搜索的迭代次数 NC , NC_{max} 是最大迭代次数。设置系统的参数 α 、 β 和 ρ 的值。

②对于人工蚂蚁 k ,其遵守下列规则来选择搜索的下一个栅格。在时刻 t ,人工蚂蚁 k 在点 p_i ,其选择下一搜索点 p_j 的概率可以用式(3-20)来计算。

$$p_{i,j}^k(t) = \begin{cases} \left[\tau_{i,j}^\alpha(t) \eta_{i,j}^\beta(t) \right] / \left\{ \sum_{\gamma \in S_i^k} [\tau_{i,\gamma}^\alpha(t) \eta_{i,\gamma}^\beta(t)] \right\}, & j \in KS_i^k \\ 0, & \text{否则} \end{cases} \quad (3-20)$$

其中 $\eta_{i,j}(t)$ 是启发式函数, 参数 α 和 β 影响 $\tau_{i,j}(t)$ 和 $\eta_{i,j}(t)$ 对 $p_{i,j}^k(t)$ 的权重。 KS_i^k 是人工蚂蚁 k 在点 p_i 能够选择的所有下一个点的集合。如果 $\eta_{i,j}(t)=1/d_{i,j}$, 则该启发式函数仅考虑距离因素, 从而导致所获得的规划路径在很多情况下有很多的转折。在本书中考虑了距离和转折两个因素来计算 $\eta_{i,j}(t)$, 见式(3-21)。

$$\eta_{i,j}(t) = \frac{h_{\text{angle}}}{d_{i,j}} \quad (3-21)$$

其中 h_{angle} 是转角因子。在本书所定义的栅格地图环境中有四种转角: 45° 、 90° 、 135° 和 180° 。转角因子 h_{angle} 值就是基于这四种转角设置的。路径转角越大则 h_{angle} 的值越小。启发式函数既考虑了距离因素又考虑了转角因素, 从而使得得到的路径在路径短的基础上转折少了。为了在算法的初期拓宽搜索的区域, 设置在初始阶段人工蚂蚁以较大的概率进行随机搜索, 而以较小的概率采用通过轮盘赌和公式(3-20)得到的概率进行下一点的选取。在算法初期, 人工蚂蚁大量拓宽搜索区域, 从而减轻了陷入局部最优解的问题。在算法中期采用通过轮盘赌和公式(3-20)得到的概率进行下一点的选取使得算法平稳进行。在算法的后期, 大多数的人工蚂蚁会以公式(3-20)算出的最大概率来进行下一点的选择, 从而加快了算法的收敛速度。

③利用公式(3-10)和公式(3-11)来更新路径上的信息素。

④在考虑转折因子的情况下计算当前所得到的最优路径。找到得到的最优路径, 并保存所有的栅格点。

⑤系统迭代次数 $NC=NC+1$ 。如果 $NC < NC_{\max}$ 则清空禁忌集合, 然后回到步骤②继续进行搜索。如果 $NC=NC_{\max}$ 则算法结束, 输出得到的最优路径。

6. 移动机器人路径规划仿真

利用 MATLAB 软件来进行移动机器人路径规划的仿真验证。仿真中移动机器人运行的环境用一个 20×20 的栅格地图来表达。分别采用 A* 算法和改进的蚁群算法来进行移动机器人路径规划。在蚁群算法中, 人工蚁群的数量设置为 $m=25$, 最大迭代次数设置为 $NC_{\max}=200$, 其他相关参数设置为: $\alpha=1$ 、 $\beta=6$ 、 $\rho=0.2$ 和 $Q=800$ 。当路径的转角分别为 0° 、 45° 、 90° 、 135° 和 180° 时, 相应的转角因子分别设置为 1、0.9、0.7、0.5 和 0.1。图 3-14 显示了利用 A* 算法所得到的最优路径, 图 3-15、图 3-16 和图 3-17 分别为采用蚁群算法所得到的最优路径。图中, 阴影部分表示的是有障碍的地点, 箭头指示的是所得到的规划路径。

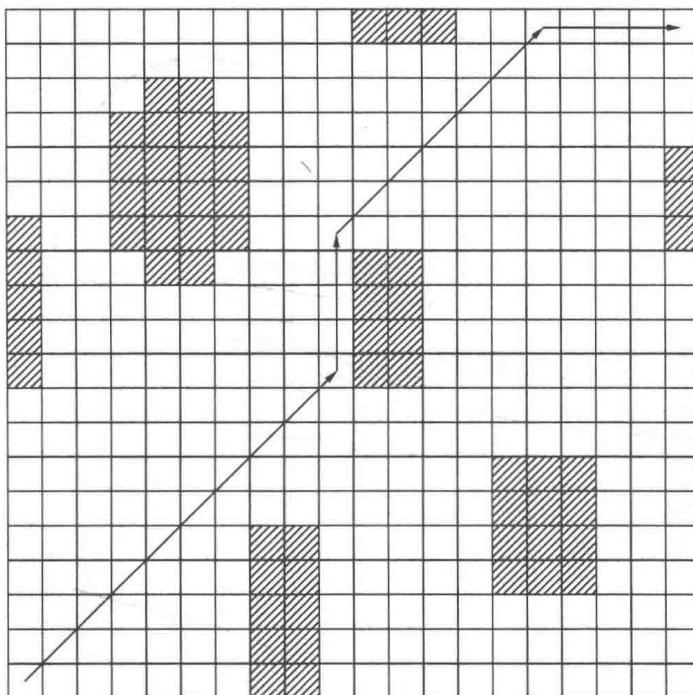


图 3-14 利用 A* 算法得到的最优路径

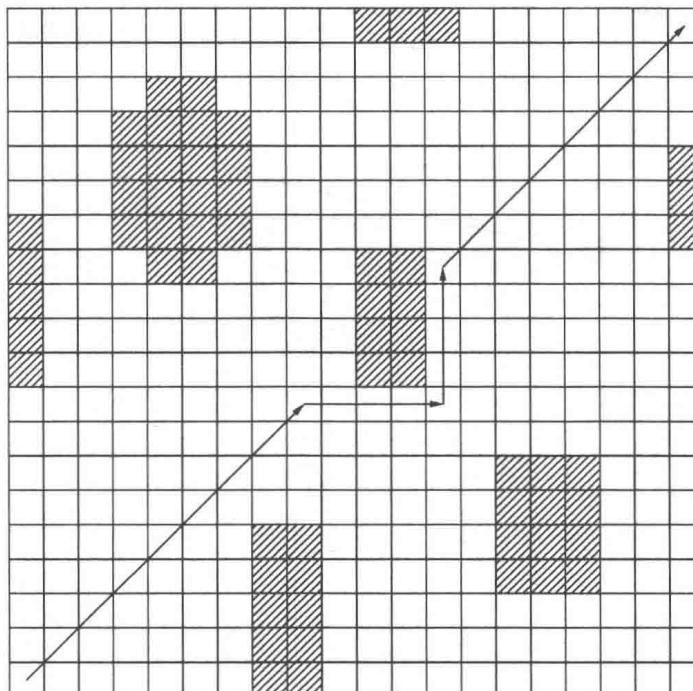


图 3-15 利用蚁群算法得到的最优路径 1

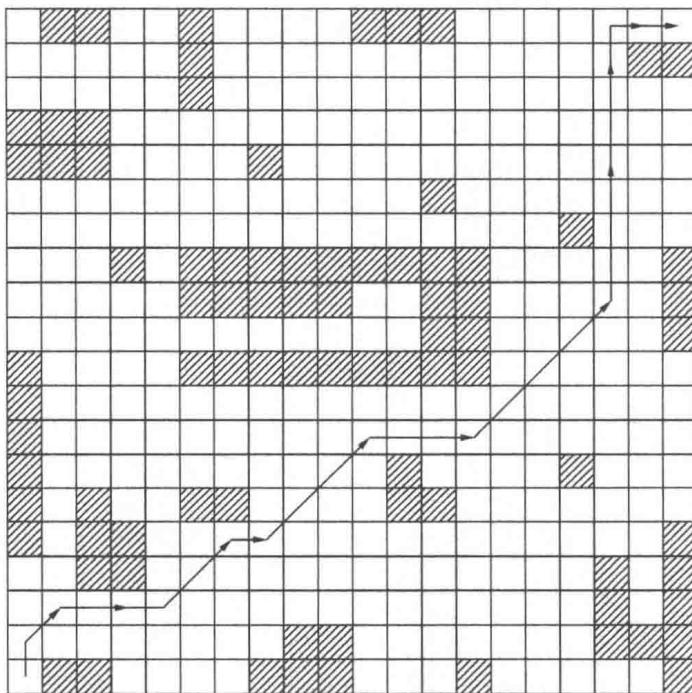


图 3-16 利用蚁群算法得到的最优路径 2

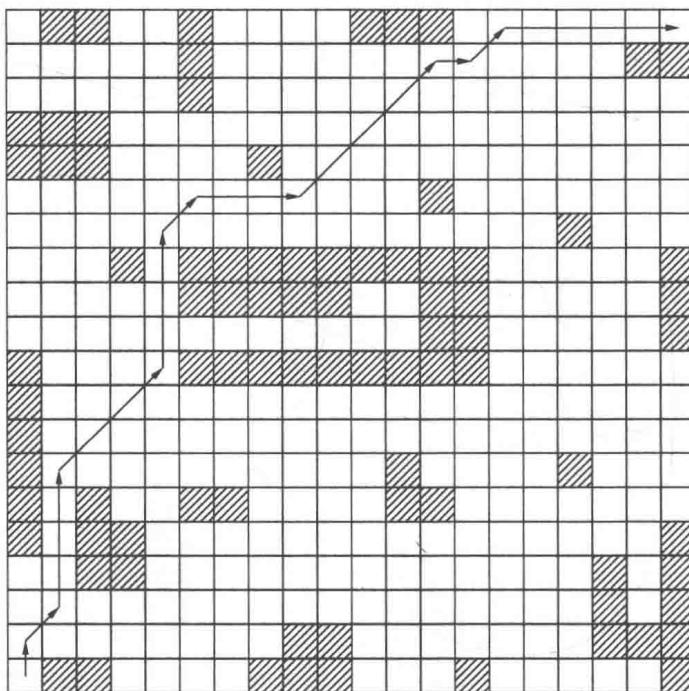


图 3-17 利用蚁群算法得到的最优路径 3

第4章 移动机器人路径跟踪控制

4.1 数学基础

远程控制系统和越来越复杂的控制算法使得控制系统中出现了不可避免的延时现象。在此采用模型算法控制(MAC)方法进行移动机器人的路径跟踪控制系统的设计。在采用GPS/DR进行导航定位解算和MAC进行控制信号解算时可能会产生时间延时。所采用的MAC方法是一种数字控制方法,因此需要将移动机器人的连续数学模型离散化,然后基于所得到的系统离散模型进行控制器的设计。本部分首先介绍非线性系统的离散问题。

1. 单输入非线性延时系统的离散

有延时的单输入单输出非线性连续系统可以用式(4-1)所示的状态空间表达式来表示。

$$\dot{x}(t) = f[x(t), u(t-D)] \quad (4-1)$$

其中, $x \in X \subset \mathbb{R}^n$ 是状态参数向量, $u \in \mathbb{R}$ 是输入信号, D 是影响输入信号的系统的恒定延时时间。

在时间坐标轴上取相等的时间间隔 $T = t_{k+1} - t_k > 0$, 其中采样时间间隔为 $[t_k, t_{k+1}] = [kT, (k+1)T]$, T 是系统的采样周期。假设公式(4-1)的输入信号在1个采样周期内是一个常数,也就是说零阶保持假设公式(4-2)成立。

$$u(t) = u(kT) \equiv u(k) = \text{常数} \quad (4-2)$$

其中 $kT \leq t < kT + T$ 。对系统输入信号延时 D 可以做如下设置^[99,100]。

$$D = qT + \gamma \quad (4-3)$$

其中, $q \in \{0, 1, 2, \dots\}$, $0 < \gamma < T$ 。也就是说系统输入信号延时 D 可以表示为整数倍数的采样周期和小于采样周期的时间的和。基于零阶保持假设和公式(4-3),可以在一个采样周期内将系统的输入信号表示如下。

$$u(t-D) = \begin{cases} u(kT - qT - T) \equiv u(k-q-1), & \text{如果 } kT \leq t < kT + \gamma \\ u(kT - qT) \equiv u(k-q), & \text{如果 } kT + \gamma \leq t < kT + T \end{cases} \quad (4-4)$$

基于零阶保持假设在一个采样周期内系统(4-1)可以采用公式(4-5)进行离散^[100]。

$$\begin{cases} x(kT + \gamma) = x(kT) + \sum_{l=1}^{\infty} A^l [x(kT), u(k-q-1)] \frac{\gamma^l}{l!}, \text{如果 } kT \leq t < kT + \gamma \\ x(kT + T) = x(kT + \gamma) + \sum_{l=1}^{\infty} A^l [x(kT + \gamma), u(k-q)] \frac{(T-\gamma)^l}{l!}, \end{cases}, \quad (4-5)$$

如果 $kT + \gamma \leq t < kT + T$

其中, $x(k)$ 是系统状态参量 x 在时刻 $t=t_k=kT$ 的值, $A^l(x, u)$ 可以用公式(4-6)进行计算。

$$\begin{cases} A^{[1]}(x, u) = f(x, u) \\ A^{[l+1]}(x, u) = \frac{\partial A^{[l]}(x, u)}{\partial x} f(x, u), l=1, 2, 3, \dots. \end{cases} \quad (4-6)$$

因此可以得到系统(4-1)的精确的离散模型, 见公式(4-7)。

$$\begin{aligned} x(k+1) = & x(k) + \sum_{l=1}^{\infty} A^l [x(k), u(k-q-1)] \frac{\gamma^l}{l!} \\ & + \sum_{l=1}^{\infty} A^l \left\{ \{x(k) + \sum_{i=1}^{\infty} A^i [x(k), u(k-q-1)] \frac{\gamma^i}{i!}\}, u(k-q) \right\} \frac{(T-\gamma)^l}{l!} \end{aligned} \quad (4-7)$$

也可以通过保留 N 阶的泰勒级数得到系统(4-1)的近似的离散模型, 见公式(4-8)。

$$\begin{aligned} x(k+1) = & x(k) + \sum_{l=1}^N A^l [x(k), u(k-q-1)] \frac{\gamma^l}{l!} \\ & + \sum_{l=1}^N A^l \left\{ \{x(k) + \sum_{i=1}^N A^i [x(k), u(k-q-1)] \frac{\gamma^i}{i!}\}, u(k-q) \right\} \frac{(T-\gamma)^l}{l!} \end{aligned} \quad (4-8)$$

2. 多输入非线性延时系统的离散

1) 两输入非线性延时系统的离散

单输入非线性延时系统的离散算法可以扩展到多输入非线性延时系统的离散中。首先考虑的是具有两输入信号的非线性系统, 如式(4-9)所示。

$$\frac{dx(t)}{dt} = f[x(t)] + u_1(t-D_1)g_1[x(t)] + u_2(t-D_2)g_2(x(t)) \quad (4-9)$$

其中 D_1 和 D_2 是相应的输入延时。基于零阶保持假设, 系统的输入信号可以用式(4-10)来表示。

$$\begin{cases} u_1(t-D_1) \rightarrow (D_1 = q_1 T + \gamma_1) \\ u_2(t-D_2) \rightarrow (D_2 = q_2 T + \gamma_2) \end{cases} \quad (4-10)$$

在一个采样周期内输入信号可以用式(4-11)来表示。

$$\begin{cases} u_1(t-D_1) = \begin{cases} u_1(kT-q_1T-T) \equiv u_1(k-q_1-1), & \text{如果 } kT \leq t < kT+\gamma_1 \\ u_1(kT-q_1T) \equiv u_1(k-q_1), & \text{如果 } kT+\gamma_1 \leq t < kT+T \end{cases} \\ u_2(t-D_2) = \begin{cases} u_2(kT-q_2T-T) \equiv u_2(k-q_2-1), & \text{如果 } kT \leq t < kT+\gamma_2 \\ u_2(kT-q_2T) \equiv u_2(k-q_2), & \text{如果 } kT+\gamma_2 \leq t < kT+T \end{cases} \end{cases} \quad (4-11)$$

为了方便起见,考虑 $\gamma_1 < \gamma_2$ 的情况,而 $\gamma_1 > \gamma_2$ 情况与 $\gamma_1 < \gamma_2$ 情况类似,在此不再赘述。

(1) 当 $kT \leq t < kT+\gamma_1$ 时,则系统的输入信号为 $u_1(t-D_1)=u_1(k-q_1-1)$ 和 $u_2(t-D_2)=u_2(k-q_2-1)$,由此可以得到在时刻 $t=kT+\gamma_1$ 系统的精确的和近似的离散模型,如式(4-12)和式(4-13)所示。

$$x(kT+\gamma_1) = x(kT) + \sum_{l=1}^{\infty} A^l [x(kT), u_1(k-q_1-1), u_2(k-q_2-1)] \frac{\gamma_1^l}{l!} \quad (4-12)$$

$$x(kT+\gamma_1) = x(kT) + \sum_{l=1}^N A^l [x(kT), u_1(k-q_1-1), u_2(k-q_2-1)] \frac{\gamma_1^l}{l!} \quad (4-13)$$

(2) 当 $kT+\gamma_1 \leq t < kT+\gamma_2$ 时,则系统的输入信号为 $u_1(t-D_1)=u_1(k-q_1)$ 和 $u_2(t-D_2)=u_2(k-q_2-1)$,由此可以得到在时刻 $t=kT+\gamma_2$ 系统的精确的和近似的离散模型,如式(4-14)和式(4-15)所示。

$$x(kT+\gamma_2) = x(kT+\gamma_1) + \sum_{l=1}^{\infty} A^l [x(kT+\gamma_1), u_1(k-q_1), u_2(k-q_2-1)] \frac{(\gamma_2-\gamma_1)^l}{l!} \quad (4-14)$$

$$x(kT+\gamma_2) = x(kT+\gamma_1) + \sum_{l=1}^N A^l [x(kT+\gamma_1), u_1(k-q_1), u_2(k-q_2-1)] \frac{(\gamma_2-\gamma_1)^l}{l!} \quad (4-15)$$

(3) 当 $kT+\gamma_2 \leq t < kT+T$ 时,则系统的输入信号为 $u_1(t-D_1)=u_1(k-q_1)$ 和 $u_2(t-D_2)=u_2(k-q_2)$,由此可以得到在时刻 $t=kT+T$ 系统的精确的和近似的离散模型,如式(4-16)和式(4-17)所示。

$$x(kT+T) = x(kT+\gamma_2) + \sum_{l=1}^{\infty} A^l [x(kT+\gamma_2), u_1(k-q_1), u_2(k-q_2)] \frac{(T-\gamma_2)^l}{l!} \quad (4-16)$$

$$x(kT+T) = x(kT+\gamma_2) + \sum_{l=1}^N A^l [x(kT+\gamma_2), u_1(k-q_1), u_2(k-q_2)] \frac{(T-\gamma_2)^l}{l!} \quad (4-17)$$

其中, $k=0, 1, 2, 3, \dots$ 。参数 $A^{[l]}(x, u)$ 可以用式(4-18)来计算。

$$\begin{aligned} A^{[1]}(x, u) &= f(x) + u_1 g_1(x) + u_2 g_2(x) \\ A^{[l+1]}(x, u) &= \frac{\partial A^{[l]}(x, u)}{\partial x} [f(x) + u_1 g_1(x) + u_2 g_2(x)] \end{aligned} \quad (4-18)$$

$l=1, 2, 3, \dots$ 。

2) 多输入非线性延时系统的离散

在得到两输入非线性延时系统离散模型的基础上,可以进一步得出多输入非线性延时系统的离散模型。多输入多输出非线性延时连续系统可以用式(4-19)来表示^[101]。

$$\begin{aligned}\frac{dx(t)}{dt} &= f[x(t)] + \sum_{i=1}^n g_i[x(t)]u_i(t-D_i) \\ &= f[x(t)] + u_1(t-D_1)g_1[x(t)] + u_2(t-D_2)g_2[x(t)] + \dots + \\ &\quad u_n(t-D_n)g_n[x(t)]\end{aligned}\quad (4-19)$$

其中, $u_i (i=1, 2, \dots, n)$ 是系统(4-19)的输入信号, $D_i (i=1, 2, \dots, n)$ 是输入信号 u_i 的延时。设参数 $q_i (i=1, 2, \dots, n)$ 是大于等于零的整数, $0 \leq \gamma_i < T (i=1, 2, \dots, n)$, 可以用式(4-20)来表示系统(4-19)的输入信号延时时间 D_i 。

$$\left\{ \begin{array}{l} D_1 = q_1 T + \gamma_1 \\ D_2 = q_2 T + \gamma_2 \\ \vdots \\ D_i = q_i T + \gamma_i \end{array} \right. \quad (4-20)$$

在每个采样周期内, 基于零阶保持假设, 系统(4-19)的输入信号可以分别用式(4-21)、式(4-22)和式(4-23)来表示。

$$u_1(t-D_1) = \begin{cases} u_1(kT - q_1 T - T) \equiv u_1(k - q_1 - 1), & \text{如果 } kT \leq t < kT + \gamma_1 \\ u_1(kT - q_1 T) \equiv u_1(k - q_1), & \text{如果 } kT + \gamma_1 \leq t < kT + T \end{cases} \quad (4-21)$$

$$u_2(t-D_2) = \begin{cases} u_2(kT - q_2 T - T) \equiv u_2(k - q_2 - 1), & \text{如果 } kT \leq t < kT + \gamma_2 \\ u_2(kT - q_2 T) \equiv u_2(k - q_2), & \text{如果 } kT + \gamma_2 \leq t < kT + T \end{cases} \quad (4-22)$$

$$\vdots$$

$$u_i(t-D_i) = \begin{cases} u_i(kT - q_i T - T) \equiv u_i(k - q_i - 1), & \text{如果 } kT \leq t < kT + \gamma_i \\ u_i(kT - q_i T) \equiv u_i(k - q_i), & \text{如果 } kT + \gamma_i \leq t < kT + T \end{cases} \quad (4-23)$$

为了简便起见, 这里假设系统输入信号延时时间参数 $\gamma_1 < \gamma_2 < \dots < \gamma_n$ 。其他的情况与这种情况相类似。

(1) 当 $kT \leq t < kT + \gamma_1$ 时, 系统的输入信号分别为 $u_1(t-D_1) = u_1(k - q_1 - 1)$, $u_2(t-D_2) = u_2(k - q_2 - 1)$, \dots , $u_n(t-D_n) = u_n(k - q_n - 1)$ 。基于零阶保持假设可以得到在时刻 $t = kT + \gamma_1$ 系统(4-19)的精确的离散模型, 如式(4-24)所示, 并且可以通过保留一定的泰勒级数阶数 N 而得到系统(4-19)的近似的离散模型, 如式(4-25)所示。

$$x(kT + \gamma_1) = x(kT) + \sum_{l=1}^{\infty} A' [x(kT), u_1(k - q_1 - 1),$$

$$u_2(k-q_2-1), \dots, u_n(k-q_n-1)] \frac{\gamma_1^l}{l!} \quad (4-24)$$

$$x(kT + \gamma_1) = x(kT) + \sum_{l=1}^N A^l [x(kT), u_1(k-q_1-1), \\ u_2(k-q_2-1), \dots, u_n(k-q_n-1)] \frac{\gamma_1^l}{l!} \quad (4-25)$$

(2) 当 $kT + \gamma_1 \leq t < kT + \gamma_2$ 时, 系统的输入信号分别为 $u_1(t-D_1) = u_1(k-q_1)$, $u_2(t-D_2) = u_2(k-q_2-1), \dots, u_n(t-D_n) = u_n(k-q_n-1)$ 。基于零阶保持假设可以得到在时刻 $t = kT + \gamma_2$ 时系统(4-19)的精确的离散模型, 如式(4-26)所示, 并且可以通过保留一定的泰勒级数阶数 N 而得到系统(4-19)的近似的离散模型, 如式(4-27)所示。

$$x(kT + \gamma_2) = x(kT + \gamma_1) + \sum_{l=1}^{\infty} A^l [x(kT + \gamma_1), u_1(k-q_1), u_2(k-q_2-1), \dots, u_n(k-q_n-1)] \frac{(\gamma_2 - \gamma_1)^l}{l!} \quad (4-26)$$

$$x(kT + \gamma_2) = x(kT + \gamma_1) + \sum_{l=1}^N A^l [x(kT + \gamma_1), u_1(k-q_1), \\ u_2(k-q_2-1), \dots, u_n(k-q_n-1)] \frac{(\gamma_2 - \gamma_1)^l}{l!} \quad (4-27)$$

(3) 当 $kT + \gamma_i \leq t < kT + \gamma_{i+1}$ ($1 \leq i \leq n-1$) 时, 系统的输入信号分别为 $u_1(t-D_1) = u_1(k-q_1), u_2(t-D_2) = u_2(k-q_2), \dots, u_i(t-D_i) = u_i(k-q_i), u_{i+1}(t-D_{i+1}) = u_{i+1}(k-q_{i+1}-1), \dots, u_n(t-D_n) = u_n(k-q_n-1)$ 。基于零阶保持假设可以得到在时刻 $t = kT + \gamma_{i+1}$ 时系统(4-19)的精确的离散模型, 如式(4-28)所示, 并且可以通过保留一定的泰勒级数阶数 N 而得到系统(4-19)的近似的离散模型, 如式(4-29)所示。

$$x(kT + \gamma_i) = x(kT + \gamma_{i-1}) + \sum_{l=1}^{\infty} A^l [x(kT + \gamma_1), u_1(k-q_1), u_2(k-q_2), \dots, u_i(k-q_i), \\ u_{i+1}(k-q_{i+1}), \dots, u_n(k-q_n-1)] \frac{(\gamma_{i+1} - \gamma_i)^l}{l!} \quad (4-28)$$

$$x(kT + \gamma_i) = x(kT + \gamma_{i-1}) + \sum_{l=1}^N A^l [x(kT + \gamma_1), u_1(k-q_1), u_2(k-q_2), \dots, u_i(k-q_i), \\ u_{i+1}(k-q_{i+1}), \dots, u_n(k-q_n-1)] \frac{(\gamma_{i+1} - \gamma_i)^l}{l!} \quad (4-29)$$

(4) 当 $kT + \gamma_n \leq t < kT + T$ 时, 系统的输入信号分别为 $u_1(t - D_1) = u_1(k - q_1)$ 、 $u_2(t - D_2) = u_2(k - q_2)$ 、 \cdots 、 $u_n(t - D_n) = u_n(k - q_n)$ 。基于零阶保持假设可以得到在时刻 $t = kT + T$ 时系统(4-19)的精确的离散模型, 如式(4-30)所示, 并且可以通过保留一定的泰勒级数阶数 N 而得到系统(4-19)的近似的离散模型, 如式(4-31)所示。

$$\begin{aligned} x(kT + T) &= x(kT + \gamma_n) \\ &+ \sum_{l=1}^{\infty} A^l [x(kT + \gamma_n), u_1(k - q_1), u_2(k - q_2), \dots, \\ &u_n(k - q_n)] \frac{(T - \gamma_n)^l}{l!} \end{aligned} \quad (4-30)$$

$$\begin{aligned} x(kT + T) &= x(kT + \gamma_n) \\ &+ \sum_{l=1}^N A^l [x(kT + \gamma_n), u_1(k - q_1), u_2(k - q_2), \dots, \\ &u_n(k - q_n)] \frac{(T - \gamma_n)^l}{l!} \end{aligned} \quad (4-31)$$

其中, $k = 0, 1, 2, 3, \dots$ 。参数 A^l 可以用公式(4-18)来计算。

式(4-24)至式(4-31)就是所得到的多输入非线性延时系统的精确离散模型和近似离散模型。

4.2 模型算法控制(MAC)

GPS/DR 综合导航系统可以为移动机器人提供长期稳定的导航信息。在基于第三章所提到的路径规划算法得到规划好的路径后, 需要设计相应的路径跟踪控制算法来控制移动机器人跟踪所设计的最优路径。本书采用模型算法控制(MAC)方法进行轮式移动机器人的路径跟踪控制算法的设计。MAC 方法是基于轮式移动机器人的模型进行设计的。因此, 首先考虑轮式移动机器人的运动学模型和动力学模型。

1. 轮式移动机器人的数学模型

为了得到轮式移动机器人的运动学模型和动力学模型, 要做如下的假设: 移动机器人的轮子与地面是纯滚动而无相对滑动的, 也就是说具有非完整运动约束的系统。因此, 基于系统的运动可以由具有 i 个约束的 j 个广义坐标 ($i < j$) 来表达, 则可以推导出具有非完整运动约束的轮式移动机器人的运动学模型和动力学模型。

考虑一个两轮驱动的非完整运动约束的轮式移动机器人, 如图 4-1 所示。

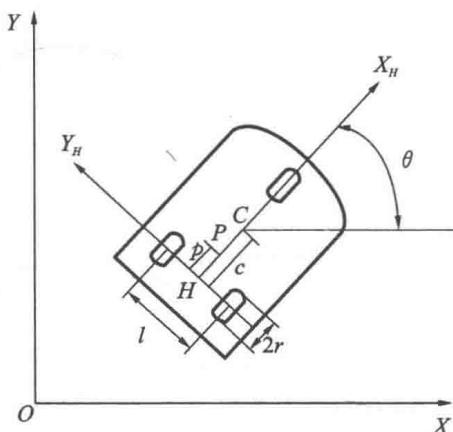


图 4-1 一个非完整运动约束的轮式移动机器人结构图

1) 轮式移动机器人运动学模型

图 4-1 所示的一个非完整运动约束的轮式移动机器人结构图给出了推导该移动机器人运动学模型的必要的信息参数。该轮式移动机器人拥有两个驱动轮和一个辅助轮, 轮子的直径为 r 。其中两个驱动轮分别由两个直流电动机独立驱动。点 $H(x_H, y_H)$ 在两个驱动轮轴的连线上, 两个驱动轮相对于该点对称, 设点 $H(x_H, y_H)$ 为参考坐标系 $\{X_H, Y_H\}$ 的原点。点 $C(x_c, y_c)$ 是该移动机器人的重心。点 P 是进行移动机器人路径跟踪控制系统设计的控制点。长度 p 是点 H 和点 P 之间的距离, 长度 c 是点 H 和点 C 之间的距离, l 是后轮轴的长度。

对于上述的轮式移动机器人, 存在着三个约束条件^[102]。第一个约束条件是移动机器人必须沿着其对称轴的方向移动, 也就是说要满足公式(4-32)。

$$\dot{x}_H \sin \theta - \dot{y}_H \cos \theta = 0 \quad (4-32)$$

另外两个约束条件是移动机器人的轮子与地面之间是纯滚动而无滑动的, 如公式(4-33)和公式(4-34)所示。

$$\dot{x}_H \cos \theta + \dot{y}_H \sin \theta + \frac{l}{2} \dot{\theta} - r \dot{\varphi}_r = 0 \quad (4-33)$$

$$\dot{x}_H \cos \theta + \dot{y}_H \sin \theta - \frac{l}{2} \dot{\theta} - r \dot{\varphi}_l = 0 \quad (4-34)$$

其中, θ 是移动机器人的运动方向和 X 轴之间的夹角, φ_r 和 φ_l 分别是左、右两个驱动轮转动的角度。

设 $q = (x_H, y_H, \theta, \varphi_r, \varphi_l)$, 则上述的三个约束条件可以写成式(4-35)的形式。

$$\mathbf{A}(q) \dot{q} = 0 \quad (4-35)$$

其中,

$$\mathbf{A}(q) = \begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 & 0 \\ \cos\theta & \sin\theta & \frac{l}{2} & -r & 0 \\ \cos\theta & \sin\theta & -\frac{l}{2} & 0 & -r \end{bmatrix} \quad (4-36)$$

设 $\mathbf{S}(q)$ 是一个由一系列光滑的、线性独立的矢量组成的满秩矩阵, 且满足式 (4-37)。

$$\mathbf{A}(q)\mathbf{S}(q)=0 \quad (4-37)$$

考虑该轮式移动机器人的运动学特性, 可以得出式(4-38)。

$$\mathbf{S}(q) = \frac{r}{2} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{2}{l} & -\frac{2}{l} \\ \frac{2}{r} & 0 \\ 0 & \frac{2}{r} \end{bmatrix} \quad (4-38)$$

设置 $\omega(t)$ 和 $v(t)$ 分别是移动机器人的线速度和角速度, 则移动机器人的运动学模型可以用式(4-39)来表示。

$$\begin{cases} \dot{x}_H = v(t) \cos(\theta) \\ \dot{y}_H = v(t) \sin(\theta) \\ \dot{\theta} = \omega(t) \end{cases} \quad (4-39)$$

式(4-39)表示的移动机器人的运动学模型是在移动机器人的轮子与地面之间是纯滚动而无滑动的假设下得到的。设 $\eta(t)$ ($|\eta(t)| \leq \rho$) 为一有界的外部扰动, 也就是说 $\eta(t)$ 是式(4-39)所示移动机器人的运动学模型违反非完整约束的量^[102]。其中 ρ 是一已知的常量。则式(4-32)可以写成式(4-40)的形式。

$$\dot{x}_H \sin\theta - \dot{y}_H \cos\theta = \eta(t) \quad (4-40)$$

于是带有外部扰动的移动机器人的运动学模型可以用式(4-41)来表示。

$$\begin{cases} \dot{x}_H = v(t) \cos(\theta) + \eta(t) \sin(\theta) \\ \dot{y}_H = v(t) \sin(\theta) - \eta(t) \cos(\theta) \\ \dot{\theta} = \omega(t) \end{cases} \quad (4-41)$$

2) 移动机器人的动力学方程

如前所述的轮式移动机器人的动力学方程可以用式(4-42)来表达^[102]。

$$\mathbf{M}(q)\ddot{\mathbf{q}} + \mathbf{V}(q, \dot{q})\dot{\mathbf{q}} = \mathbf{E}(q)\boldsymbol{\tau} - \mathbf{A}^T(q)\boldsymbol{\lambda} \quad (4-42)$$

其中, $\mathbf{M}(q) \in \mathbf{R}^{j \times j}$ 是一个对称的正定惯性矩阵, $\mathbf{V}(q, \dot{q}) \in \mathbf{R}^{j \times j}$ 是向心力和科里奥利力矩阵, $\mathbf{A}(q) \in \mathbf{R}^{i \times j}$ 是与约束条件有关的矩阵, $\boldsymbol{\lambda} \in \mathbf{R}^i$ 是拉格朗日算子向量, $\mathbf{E}(q) \in \mathbf{R}^{j \times p}$ 是输入变换矩阵, $\boldsymbol{\tau} \in \mathbf{R}^p$ 是输入向量, 也就是施加在两个驱动轮上的驱动力矩。

式(4-43)和式(4-44)可以用来表示式(4-42)所示的非完整约束移动机器人的动力学方程。

$$\dot{\mathbf{q}} = \mathbf{S}(q)\boldsymbol{\eta}(t) \quad (4-43)$$

$$\overline{\mathbf{M}}\dot{\mathbf{q}} + \overline{\mathbf{V}}\dot{\mathbf{q}} = \overline{\mathbf{B}}\boldsymbol{\tau} \quad (4-44)$$

其中

$$\overline{\mathbf{M}}(q) = \begin{bmatrix} \frac{ml^2}{4} + I & \frac{ml^2}{4} - I \\ r^2 \frac{\frac{ml^2}{4} + I_w}{l^2} & r^2 \frac{\frac{ml^2}{4} + I_w}{l^2} \\ r^2 \frac{\frac{ml^2}{4} - I}{l^2} & r^2 \frac{\frac{ml^2}{4} - I}{l^2} + I_w \end{bmatrix} \quad (4-45)$$

$$\overline{\mathbf{V}} = \begin{bmatrix} 0 & \frac{r^2 m_c c \dot{\theta}}{l} \\ -\frac{r^2 m_c c \dot{\theta}}{l} & 0 \end{bmatrix} \quad (4-46)$$

$$\overline{\mathbf{B}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4-47)$$

上述公式中, $\boldsymbol{\tau} = [\tau_r \ \tau_l]^T$ 是移动机器人的输入信号, 也就是施加在左右两个驱动轮上的力矩; $\boldsymbol{\eta} = [\eta_r \ \eta_l]^T$ 是左右两个驱动轮的角速度向量; $I = m_c c^2 + \frac{m_w l^2}{2} + I_c + 2I_m$; $m = m_c + 2m_w$ 。其中 m_c 和 m_w 分别是移动机器人平台和一个驱动轮的质量。 I_c 、 I_w 和 I_m 分别是移动机器人平台相对于通过点 H 的垂直轴的转动惯量、移动机器人的驱动轮相对于轮轴的转动惯量和移动机器人的轮子相对于轮子直径的转动惯量。

设移动机器人在点 H 的线速度和角速度的值分别为 v 和 w 。因此可以得到如式(4-48)所示的关系。

$$\begin{bmatrix} \rho_r \\ \rho_l \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{l}{2r} \\ \frac{1}{r} & -\frac{l}{2r} \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (4-48)$$

$$\dot{q} = \begin{bmatrix} \dot{x}_H \\ \dot{y}_H \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

通过以上所述,可以得到移动机器人上点 P 的动力学方程,如式(4-49)所示。

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v\cos\theta - pw\sin\theta \\ v\sin\theta + pw\cos\theta \\ w \\ \beta_1 w^2 \\ \beta_2 vw \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4-49)$$

其中, $\beta_1 = \frac{m_c cr^2}{\Theta_u}$, $\beta_2 = -\frac{2m_c cr^2}{\Theta_w}$, $\alpha_1 = \frac{r}{\Theta_u}$, $\alpha_2 = \frac{rl}{\Theta_w}$, $\Theta_u = mr^2 + 2I_w$, $\Theta_w = 2r^2 I + I_w l^2$, $u_{1,2} = (\tau_r \pm \tau_l)$ 。本书中用来进行移动机器人路径跟踪控制算法仿真和实验用机器人的各项参数值如表 4-1 所示。

表 4-1 移动机器人路径跟踪控制算法仿真和实验用机器人的各项参数值

m_c/kg	m_w/kg	$I/(\text{kg} \cdot \text{m}^2)$	$I_w/(\text{kg} \cdot \text{m}^2)$
5	0.05	0.16	0.000016
r/m	l/m	c/m	p/m
0.15	0.5	0.25	0.15

2. 线性系统的模型算法控制方法

考虑一个线性系统的离散模型,如式(4-50)所示。

$$\begin{cases} x_m(k+1) = Ax_m(k) + Bu(k) \\ y_m(k) = Cx_m(k) \end{cases} \quad (4-50)$$

其中, x_m 和 y_m 分别是系统参数 x 和 y 的预估值,该预估值是在输入信号为 $u(k)$ 时通过对系统模型进行仿真而得到的。该下标可以用来区别 y_m 和 y ,其中 y 是实际系统的测量输出。设 r 是系统的相对阶数。利用式(4-50)所示的系统模型进行仿真,可以得到系统未来输出的预测值。具体来说,基于式(4-50)所示的模型,可以得到如式(4-51)所示的系统未来输出的预测值。

$$y_m(k) = Cx_m(k)$$

$$\begin{aligned}
 y_m(k+1) &= CAx_m(k) \\
 &\vdots \\
 y_m(k+r-1) &= CA^{r-1}x_m(k) \\
 y_m(k+r) &= CA^r x_m(k) + CA^{r-1}Bu(k)
 \end{aligned} \tag{4-51}$$

基于式(4-51)可以计算出系统未来输出变化的预测值,如式(4-52)所示。

$$\begin{aligned}
 y_m(k+1) - y_m(k) &= (CA - C)x_m(k) \\
 y_m(k+2) - y_m(k) &= (CA^2 - C)x_m(k) \\
 &\vdots \\
 y_m(k+r-1) - y_m(k) &= (CA^{r-1} - C)x_m(k) \\
 y_m(k+r) - y_m(k) &= (CA^r - C)x_m(k) + CA^{r-1}Bu(k)
 \end{aligned} \tag{4-52}$$

当用式(4-52)所示的系统未来输出变化的预测值与实际系统输出的测量值 y 相加时,可以得到系统未来输出的预测值,如式(4-53)所示。

$$\begin{aligned}
 \hat{y}(k+1) &= y(k) + (CA - C)x_m(k) \\
 \hat{y}(k+2) &= y(k) + (CA^2 - C)x_m(k) \\
 &\vdots \\
 \hat{y}(k+r-1) &= y(k) + (CA^{r-1} - C)x_m(k) \\
 \hat{y}(k+r) &= y(k) + (CA^r - C)x_m(k) + CA^{r-1}Bu(k)
 \end{aligned} \tag{4-53}$$

其中,上标 $\hat{\cdot}$ 用于表示 \hat{y} 是系统输出的预测值。

基于上述内容,下一步应该是寻找合适的系统输入 $u(k)$,使得系统在 r 倍的采样周期后能够输出期望的输出值。如果所选的系统输入 $u(k)$ 使得 $\hat{y}(k+r)$ 能够等于实际的设定值 $y_{sp}(k)$,则会使得实际的系统的鲁棒性变差。为了提高系统的鲁棒性,可以使得 $\hat{y}(k+r)$ 能够沿着逼近实际的设定值 $y_{sp}(k)$ 的方向,且与 $\hat{y}(k+r-1)$ 和实际的设定值 $y_{sp}(k)$ 保持一定的距离。也就是说,可以定义一个在 $(k+r)$ 采样周期的系统输出期望值 y_d ,如式(4-54)所示。

$$y_d(k+r) = (1-\alpha)y_{sp}(k) + \alpha\hat{y}(k+r-1) \tag{4-54}$$

其中, $\alpha(0 < \alpha < 1)$ 是一个可以调整的参数。从式(4-54)可以容易看出,当 $\alpha \rightarrow 0$ 时, $y_d(k+r) \rightarrow y_{sp}(k)$ 。这个时候系统的输入会使得系统的输出值能够以最快的速度逼近系统的给定值。而当 $\alpha \rightarrow 1$ 时, $y_d(k+r) \rightarrow \hat{y}(k+r-1)$ 。这种情况下系统的输入对系统的输出无影响。选取一个合适的 α 的中间值可以使得系统的输出介于 $y_{sp}(k)$ 和 $\hat{y}(k+r-1)$ 之间,使得系统的输出能够以正确的方向逼近系统的设定值。式(4-54)通常被称为参考轨迹。

一旦选定了系统输出期望值 y_d ,下一个问题是如何得到系统的控制输入 $u(k)$,使得 $\hat{y}(k+r)$ 能够等于 $y_d(k+r)$ 。这个问题是一个求解优化的问题,如式(4-55)所示。

$$\min_{u(k)} [y_d(k+r) - \hat{y}(k+r)]^2 \tag{4-55}$$

基于式(4-52)和式(4-53),式(4-55)可以写成如式(4-56)的形式。

$$\min_{u(k)} \{(1-\alpha)e(k) - [(CA^r - C) - \alpha(CA^{r-1} - C)]x_m(k) - CA^{r-1}Bu(k)\}^2 \quad (4-56)$$

其中, $e(k) = y_{sp}(k) - y(k)$ 。

3. 非线性系统的模型算法控制(MAC)方法

线性系统的模型算法控制方法可以推广到非线性系统中。考虑一个非线性系统的离散模型,如式(4-57)所示。

$$\begin{cases} x_m(k+1) = \Phi[x_m(k), u(k)] \\ y_m(k) = h[x_m(k)] \end{cases} \quad (4-57)$$

其中,下标 m 用来表示基于式(4-57)所示的系统模型仿真所得到的 x 和 y 的预估值,将 y_m 和实际系统输出的测量值 y 区分开。

设式(4-57)的相对阶数为 r ,也就是说式(4-58)成立。

$$\begin{cases} \frac{\partial h(x)}{\partial x} \left[\frac{\partial \Phi(x, u)}{\partial x} \right]^l \frac{\partial \Phi(x, u)}{\partial u} = 0 \\ l = 0, 1, \dots, r-2 \\ \frac{\partial h(x)}{\partial x} \left[\frac{\partial \Phi(x, u)}{\partial x} \right]^{r-1} \frac{\partial \Phi(x, u)}{\partial u} \neq 0 \end{cases} \quad (4-58)$$

类似于线性系统模型算法控制方法,对式(4-57)所示的系统模型进行仿真可以得到系统输出未来变化的预测值,如式(4-59)所示。

$$\begin{aligned} y_m(k+1) - y_m(k) &= h^1[x_m(k)] - h[x_m(k)] \\ y_m(k+2) - y_m(k) &= h^2[x_m(k)] - h[x_m(k)] \\ &\vdots \\ y_m(k+r-1) - y_m(k) &= h^{r-1}[x_m(k)] - h[x_m(k)] \\ y_m(k+r) - y_m(k) &= h^{r-1}\{\Phi[x_m(k), u(k)]\} - h[x_m(k)] \end{aligned} \quad (4-59)$$

其中, r 是式(4-57)的相对阶数,并且做如式(4-60)所示的设置。

$$\begin{cases} h^0(x) = h(x) \\ h^l(x) = h^{l-1}[\Phi(x, u)], l = 1, 2, \dots, r-1 \end{cases} \quad (4-60)$$

式(4-58)可以进一步表示成式(4-61)的形式。

$$\begin{cases} \frac{\partial h^l[\Phi(x, u)]}{\partial u} = 0, l = 0, 1, \dots, r-2 \\ \frac{\partial h^{r-1}[\Phi(x, u)]}{\partial u} \neq 0 \end{cases} \quad (4-61)$$

并且,基于前述内容可以得到式(4-62)所示关系。

$$\begin{cases} y(k+l) = h^l[x(k)] \\ y(k+r) = h^{r-1}\{\Phi[x(k), u(k)]\} \\ l = 0, 1, \dots, r-1 \end{cases} \quad (4-62)$$

其中, r 是系统的相对阶数, 也就是说需要经过至少 r 个采样周期后系统输入信号 u 才能够影响系统的输出 y 。如果系统的相对阶数 r 是一个有限的数, 则 $\frac{\partial h^{r-1}(\Phi(x, u))}{\partial u} \neq 0$ 意味着式(4-63)是有解的。

$$h^{r-1}[\Phi(x, u)] = y \quad (4-63)$$

为了叙述方便, 做如式(4-64)所示的设置。

$$u = \Psi_0(x, y) \quad (4-64)$$

并且设式(4-64)中系统输入 u 在 $X \times h(X)$ 上为唯一解。

将式(4-59)所示的系统输出未来变化的预测值加上系统实际输出的测量值 y , 可以得到如式(4-65)所示的系统输出的预估值。

$$\begin{aligned} \hat{y}(k+1) &= y(k) + h^1[x_m(k)] - h[x_m(k)] \\ \hat{y}(k+2) &= y(k) + h^2[x_m(k)] - h[x_m(k)] \\ &\vdots \\ \hat{y}(k+r-1) &= y(k) + h^{r-1}[x_m(k)] - h[x_m(k)] \\ \hat{y}(k+r) &= y(k) + h^{r-1}\{\Phi[x_m(k), u(k)]\} - h[x_m(k)] \end{aligned} \quad (4-65)$$

从式(4-65)可以看出系统的输入信号 u 在 r 个采样周期后才对系统的输出有影响, 这与前面提到的系统总共有 $r\Delta t$ 的延时时间相符合, 其中 $(r-1)\Delta t$ 是系统的死区时间, Δt 是系统采样延时时间。

类似于式(4-54)所示的线性系统情况下的系统参考轨迹, 定义相应的非线性模型参考控制系统的参考轨迹, 基于此参考轨迹可以进行非线性模型参考控制算法的设计, 以控制系统的输出能够跟踪所定义的参考轨迹。具体来说就是求解式(4-66)的最小值。

$$\min_{u(k)} [y_d(k+r) - \hat{y}(k+r)]^2 \quad (4-66)$$

考虑式(4-55)和式(4-61), 式(4-66)可以写成式(4-67)的形式。

$$\min_{u(k)} \{(1-\alpha)e(k) - h^{r-1}\{\Phi[x_m(k), u(k)]\} + \alpha h^{r-1}[x_m(k)] + (1-\alpha)h[x_m(k)]\}^2 \quad (4-67)$$

其中, $e(k) = y_{sp}(k) - y(k)$ 。

在不考虑存在输入约束的情况下, 式(4-67)所示的求最小值的问题是有解的。也就是说系统的输入 $u(k)$ 是式(4-68)所示的非线性代数方程的解。

$$h^{r-1}\{\Phi[x_m(k), u(k)]\} = b[x_m(k), e(k)] \quad (4-68)$$

其中, $b(x, e) = \alpha h^{r-1}(x) + (1-\alpha)(h(x) + e)$ 。

考虑式(4-64)中 Ψ_0 的定义,式(4-68)所示的非线性代数方程的解可以表示为如式(4-69)所示的形式。

$$u(k) = \Psi_0 \{ x_m(k), b[x_m(k), e(k)] \} \quad (4-69)$$

因此,式(4-69)所示的 $u(k)$ 的值就是所设计的控制系统的控制输入,其中 x_m 可以通过对 $x_m(k+1) = \Phi[x_m(k), u(k)]$ 进行仿真而得到。

4. 非线性模型算法控制算法仿真

通过将前述的非线性模型算法控制方法应用到两个非线性延时系统进行仿真来验证所提出的非线性模型算法控制方法的有效性。

1) 简单的非线性连续系统

考虑一个简单的非线性连续系统,如式(4-70)所示^[103]。

$$\begin{aligned} \dot{x}(t) &= \frac{1 - e^{-x(t)}}{1 + e^{-x(t)}} + u(t) \\ y(t) &= x(t) \end{aligned} \quad (4-70)$$

首先考虑系统存在小延时的情况。在仿真中系统的初始值设为 $x(0)=0$,系统的采样周期为 $T=0.05$ s,延时的时间为 $D=0.02$ s。通过采用如前所述的非线性延时系统离散方法,取泰勒级数的阶数为 $N=2$,可以得到如式(4-71)所示的系统(4-70)的离散模型。

$$\begin{aligned} x(kT+D) &= x(kT) + 0.02A(kT) + 0.02u(kT-T) \\ &\quad + 0.0002\{\exp[x(kT)]B(kT) \\ &\quad + A(kT)B(kT)\exp[x(kT)]\}[A(kT) + u(kT-T)] \\ x(kT+T) &= x(kT+D) + 0.02A(kT+D) + 0.02u(kT) \\ &\quad + 0.0002\{\exp[x(kT+D)]B(kT+D) \\ &\quad + A(kT+D)B(kT+D)\exp[x(kT+D)]\}[A(kT+D) + u(k)] \end{aligned} \quad (4-71)$$

$$\text{其中, } A(k) = \frac{1 - \exp[-x(k)]}{1 + \exp[-x(k)]}, B(k) = \frac{1}{1 + \exp[-x(k)]}$$

在这种情况下 $r=1$,也就是说一个采样周期后系统的输入 u 可以影响到系统的输出 y 。在仿真中,模型算法控制方法的可调滤波参数 α ,选为 $\alpha=0.8$ 。图 4-2 展示了系统的给定值和被控对象的实际输出值。图 4-3 展示了控制系统的输入值。可以采用不同的模型算法控制方法的可调滤波参数 α 值来改变控制系统的性能。在后续的仿真中被控系统的输出加入了呈正态分布的噪声,其协方差为 0.02。表 4-2 展示了在选取不同的模型算法控制方法的可调滤波参数 α 的值情况下被控系统的输出误差。图 4-4 展示了表 4-2 所示的系统输出误差。

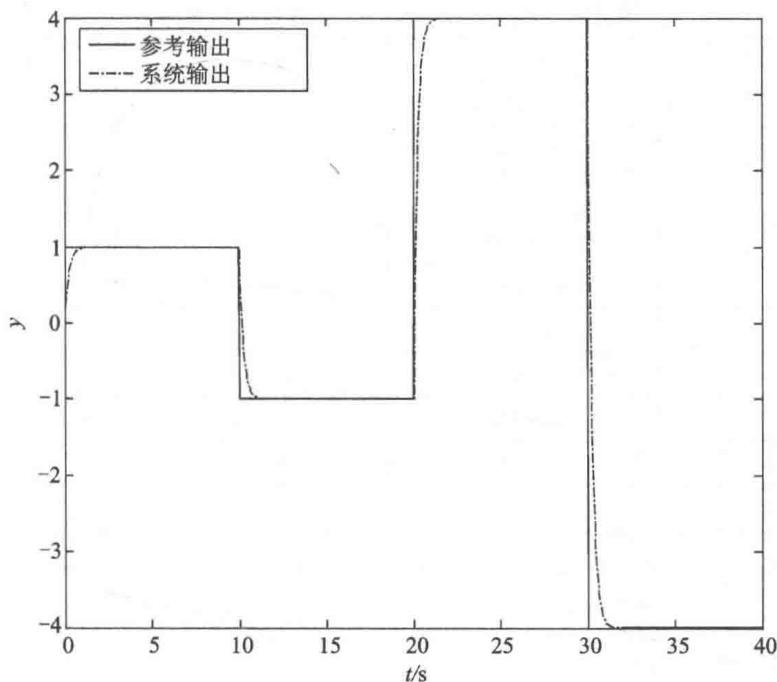
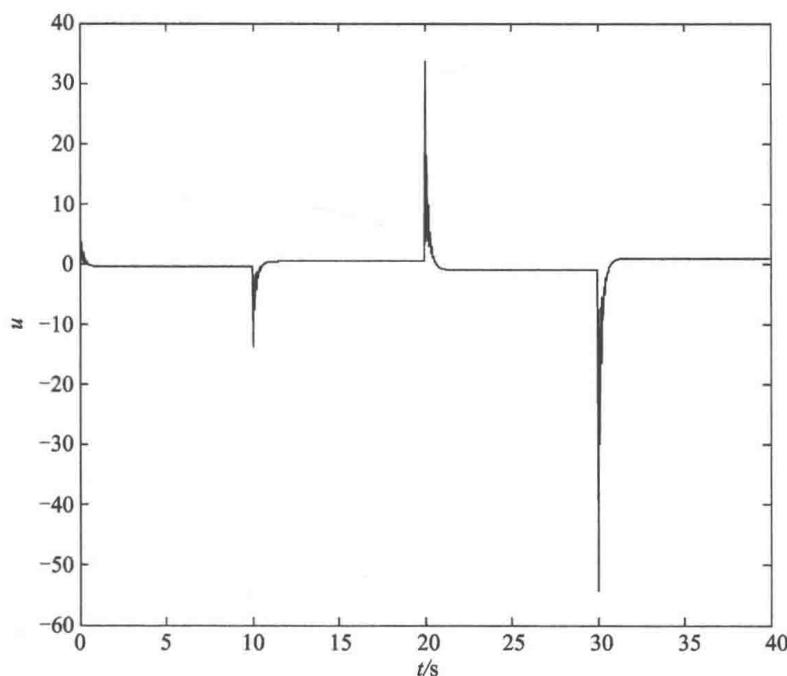
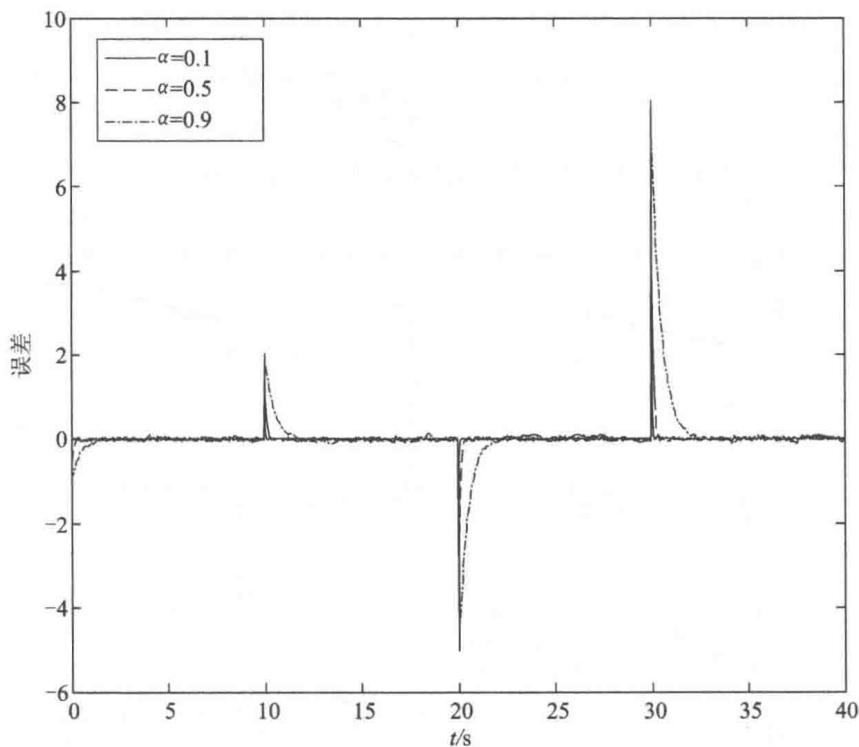
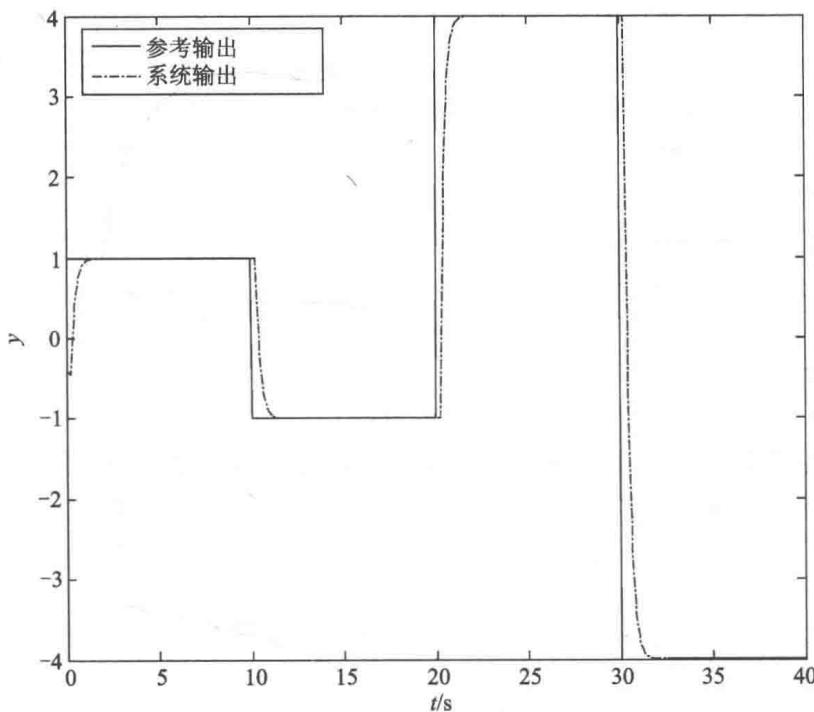
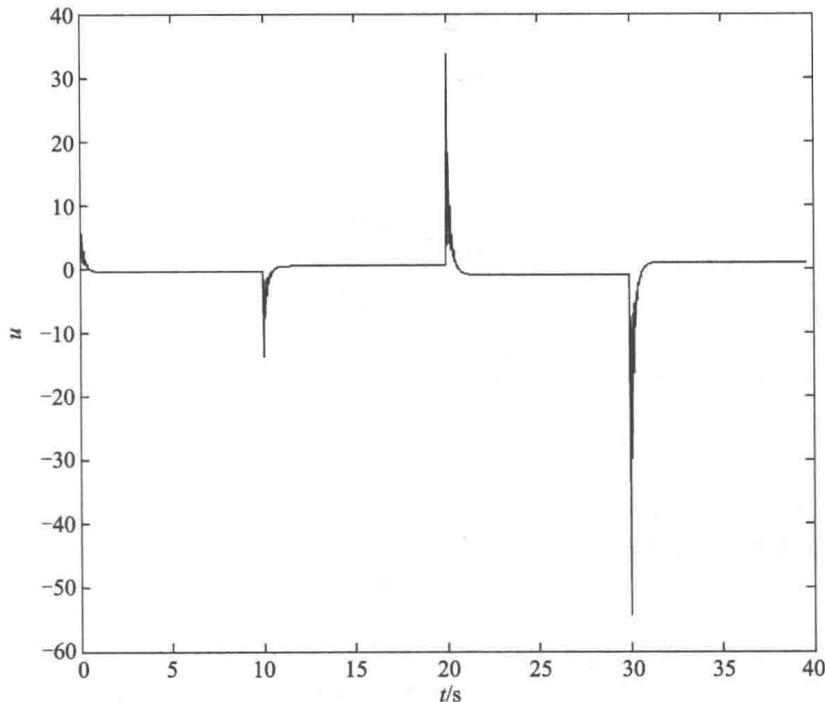
图 4-2 模型算法控制系统输出 ($D=0.02$ s, $\alpha=0.8$)图 4-3 系统控制输入 ($D=0.02$ s, $\alpha=0.8$)

表 4-2 不同 α 的值情况下系统输出的误差

采样周期	$\alpha=0.1$	$\alpha=0.5$	$\alpha=0.9$
80	-0.0022	-0.0199	0.0408
160	-0.0335	-0.0059	0.0132
240	-0.0158	-0.0383	0.0108
320	0.0161	0.0451	-0.0082
400	0.0070	-0.0141	-0.0199
480	0.0182	-0.0134	0.1132
560	-0.0090	-0.0270	-0.0321
640	0.0110	-0.0172	0.0885
720	0.0176	0.0044	0.0230
800	0.0241	0.0033	-0.0331

图 4-4 不同 α 的值情况下系统输出的误差

接下来在仿真中加大系统的延时时间, 取 $D=0.32$ s。在这种情况下 $r=6$ 。在该仿真中系统的初始值取为 $x(0)=-0.4$, 模型算法控制方法的可调滤波参数 α 选为 $\alpha=0.8$ 。图 4-5 展示了系统给定值和被控系统的输出值。图 4-6 展示了控制系统的输入。

图 4-5 模型算法控制系统输出 ($D=0.32$ s, $\alpha=0.8$)图 4-6 系统控制输入 ($D=0.32$ s, $\alpha=0.8$)

从上述仿真结果可以看出,当选择较小的模型算法控制方法的可调滤波参数 α 时,控制系统的输出将快速地跟踪系统输出的设定值,但是这种情况下系统的鲁棒性会相应地变差;另一方面,如果选择较大的模型算法控制方法的可调滤波参数 α 来提高系统的鲁棒性,则被控系统的输出会以较慢的速度跟踪系统输出的设定值。

2) 一个化工过程系统

第二个要来进行仿真的系统考虑一个化工过程系统,该系统是一个在恒温情况下工作的完全混合连续搅拌釜反应器^[104]。反应



在反应器中发生,其相应的反应率为

$$r_A = -2kC_A^2 \quad (4-73)$$

其中, k 是系统的反应率,是个常数; C_A 是反应物 A 在反应器中的浓度。

根据反应物 A 的材料平衡可以得到如式(4-74)所示的系统的非线性模型。

$$V \frac{dC_A}{dt} = FC_{Ain} - FC_A - 2kC_A^2 \quad (4-74)$$

其中, V 是反应器的容量, C_{Ain} 是向反应器中注入的反应物 A 的浓度, F 是注入反应物的流速。

为了方便起见,做如式(4-75)所示的无量纲变量的定义。

$$\left\{ \begin{array}{l} x = \frac{C_A - C_{As}}{C_{As}} \\ u = \frac{F - F_s}{F_s} \\ r = \frac{tF_s}{V} \\ a = \frac{2kC_{As}V}{F_s} = \frac{C_{Ain} - C_{As}}{C_{As}} \end{array} \right. \quad (4-75)$$

其中,下标 s 用于表示该系统处于一个稳定的状态下。

变量 u 是系统的控制输入,是注入反应物的流速 F 相对于其参考设定值的比例。基于如上所示内容系统(4-74)可以表示成式(4-76)所示的形式。

$$\frac{dx}{dr} = -(1+2a)x + au - ux - ax^2 \quad (4-76)$$

基于式(4-76),设 $a=1$,可以得到该恒温状态下的完全混合连续搅拌釜反应器化工过程的模型,如式(4-77)所示。

$$\dot{x}(t) = -x^2(t) - 3x(t) + u(t-D)[1-x(t)] \quad (4-77)$$

仿真中系统的初始值设为 $x(0)=0$ 。系统的采样周期设为 $T=0.05$ s,系统的延时时间为 $D=0.05$ s。在这种情况下取如前所述的非线性延时系统离散方法的泰勒级数的阶数为 $N=1$ 。采用如前所述的非线性延时系统离散方法可以得到系统(4-77)的离散模型,如式(4-78)所示。

$$x(k+1) = x(k) - 0.05x^2(k) - 0.15x(k) + 0.05u(k-1)[1-x(k)] \quad (4-78)$$

由于在进行非线性延时系统离散时选取的泰勒级数的阶数为 $N=1$ ，因此所得到的系统离散模型并不是很准确。采用非线性模型算法控制方法进行该非线性系统控制器的设计。在仿真中模型算法控制方法的可调滤波参数 α 选为 $\alpha=0.8$ 。系统的输出设定值为 0.8。图 4-7 展示了采用非线性模型算法控制方法的恒温情况下完全混合连续搅拌釜反应器的控制结果，图 4-8 展示了控制的输入。

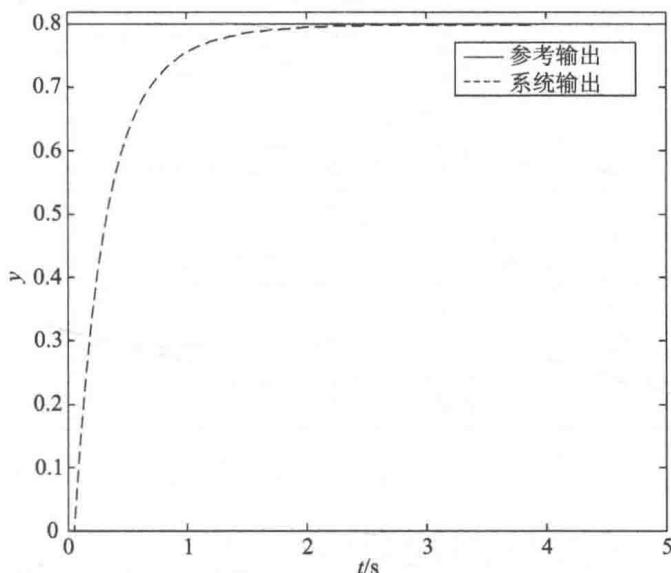


图 4-7 模型算法控制系统输出

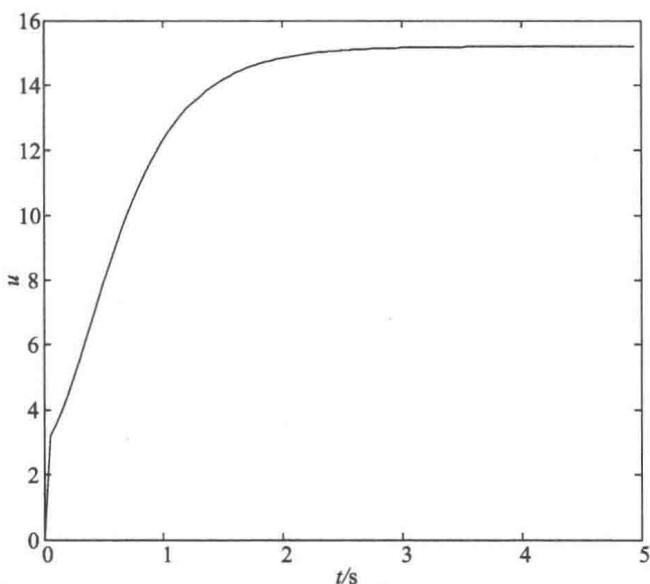


图 4-8 系统控制输入

最后,在仿真中的系统输出中加入了协方差为 0.01 的正态分布噪声。表 4-3 和图 4-9 展示了加入协方差为 0.01 的正态分布噪声后,在选用不同的模型算法控制方法的可调滤波参数 α 的值的情况下,控制系统输出的误差。

表 4-3 不同模型算法控制方法的可调滤波参数 α 的值的情况下系统输出的误差

采样周期	$\alpha=0.1$	$\alpha=0.5$	$\alpha=0.9$
10	-0.0075	-0.0412	-0.3789
20	0.0010	-0.0102	-0.1830
30	0.0243	0.0373	-0.0696
40	-0.0123	0.0208	0.0016
50	-0.0031	-0.0199	-0.0241
60	-0.0017	0.0279	-0.0221
70	0.0058	-0.0002	-0.0408
80	0.0015	0.0019	-0.0322
90	-0.0006	0.0072	-0.0324
100	0.0237	0.0012	-0.0196

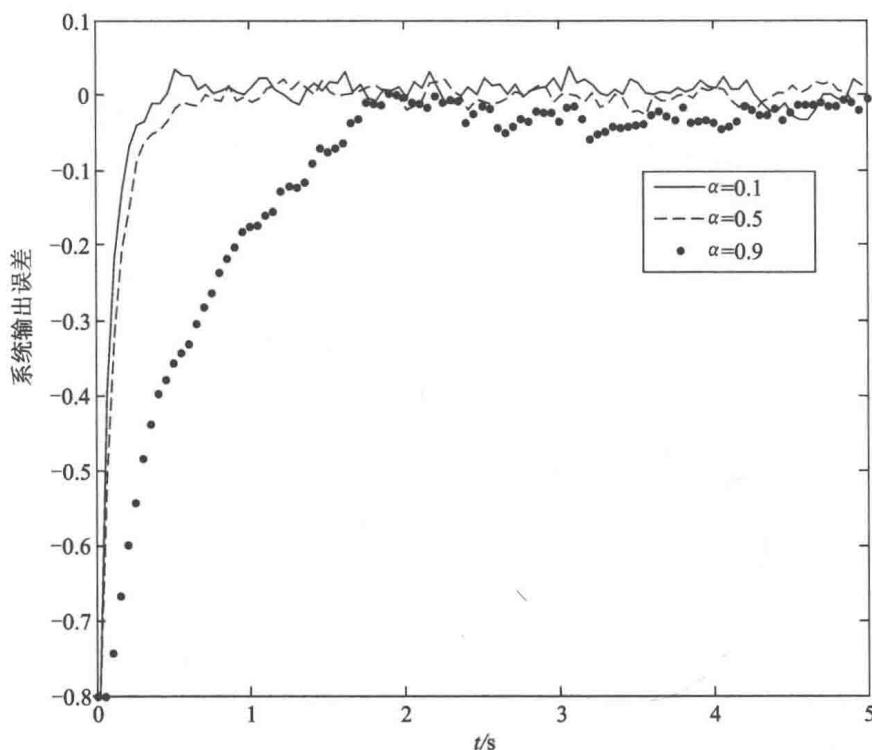


图 4-9 不同模型算法控制方法的可调滤波参数 α 的值的情况下系统输出的误差

5. 移动机器人路径跟踪非线性模型算法控制

1) 基于移动机器人运动学模型的路径跟踪控制

如前所述的非线性模型算法控制方法可以用来进行轮式移动机器人路径跟踪控制系统的设计。首先基于移动机器人运动学模型进行控制器的设计。利用前述的非线性离散算法对式(4-36)所示的轮式移动机器人运动学模型进行离散,得到其离散模型,如式(4-79)所示。

$$\mathbf{X}(k+1) = \Phi[\mathbf{X}(k), \mathbf{U}(k), \eta(k)] \quad (4-79)$$

其中, $\mathbf{X} = [x_H \ y_H \ \theta]^T$, $\mathbf{U} = [v \ \omega]^T$, $|\eta| \leq \rho$ 。然后利用式(4-69)可以得到其控制输入。

在进行移动机器人路径跟踪实验验证前,先进行基于轮式移动机器人的运动学模型仿真验证。仿真中的采样周期为 $T=0.1$ s。

首先,考虑机器人轮子与地面是纯滚动而无滑动的情况。在这种情况下 $\eta(t)=0$ 。考虑控制移动机器人跟踪一“8”字形路径。机器人的起始点坐标为 $[0.0, 0.0, 30.0^\circ]^T$ 。图 4-10 展示了参考路径和采用模型算法控制方法的移动机器人路径跟踪控制结果,图 4-10 中参考路径与跟踪路径已近似重合。图 4-11 展示了控制信号。图 4-12 展示了移动机器人的控制位置误差 $e_k = \sqrt{(e_k^x)^2 + (e_k^y)^2}$ 和方向角度误差 e_k^θ 。

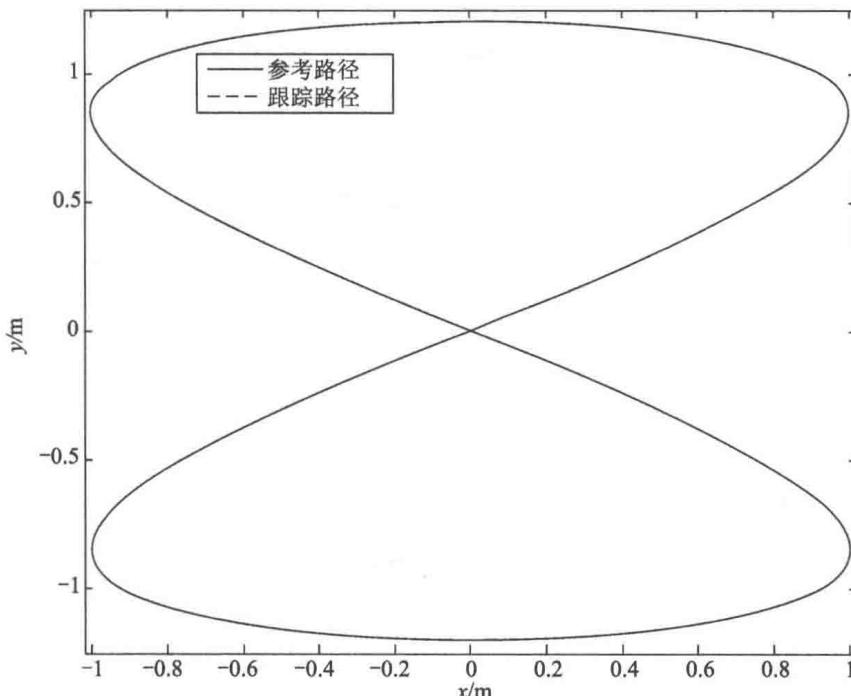
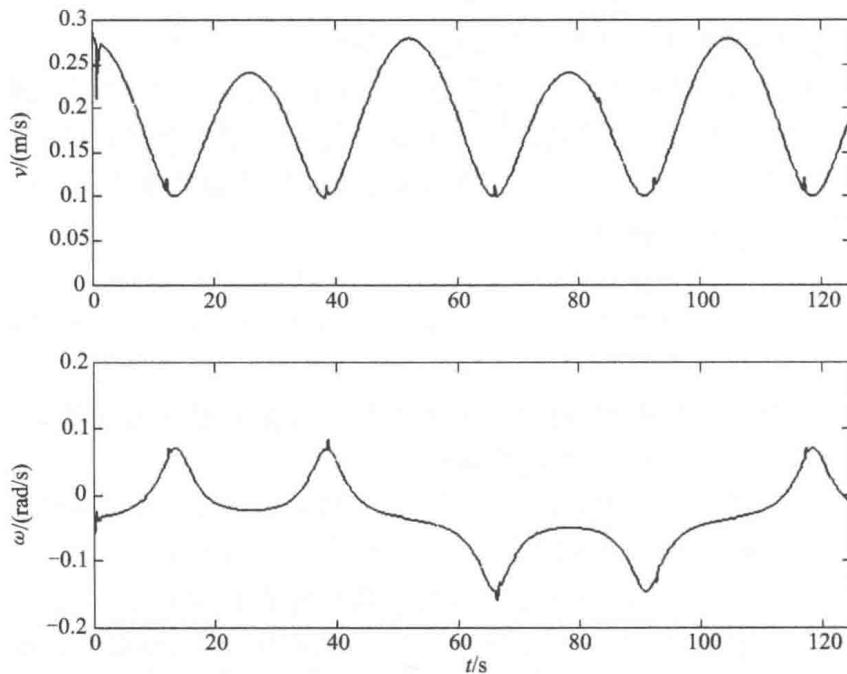
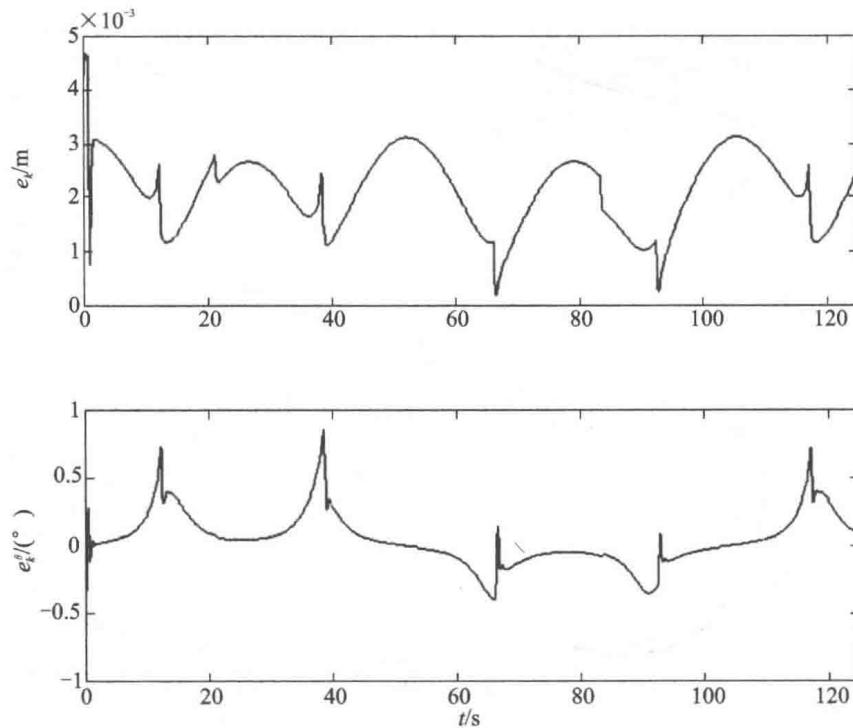


图 4-10 模型算法控制方法的路径跟踪控制仿真结果 ($\eta(t)=0$)

图 4-11 系统控制信号 ($\eta(t)=0$)图 4-12 控制位置误差 $e_k = \sqrt{(e_k^x)^2 + (e_k^y)^2}$ 和方向角度误差 e_k^θ [$\eta(t)=0$]

其次,考虑机器人的轮子与地面存在滑动的情况。在这种情况下 $\eta(t) \neq 0$ 。在仿真中所采用的扰动 $\eta(t)$ 展示在图 4-13 中。在该仿真中仍然控制移动机器人跟踪一“8”字形参考路径。移动机器人的初始位置坐标为 $[0.0, 0.0, 30.0^\circ]^T$ 。图 4-14 展示了参考路径和采用模型算法控制方法的移动机器人路径跟踪控制结果。图 4-15 展示了控制信号。图 4-16 展示了移动机器人的控制位置误差 $e_k = \sqrt{(e_k^x)^2 + (e_k^y)^2}$ 和方向角度误差 e_k^θ 。仿真中总的计算时间为 0.31 s。从图 4-14 可以看出,基于所提出的非线性模型算法控制方法设计的轮式移动机器人运动学路径跟踪控制系统,可以在轮子与地面存在滑动的情况下,为移动机器人提供很好的路径跟踪控制效果。

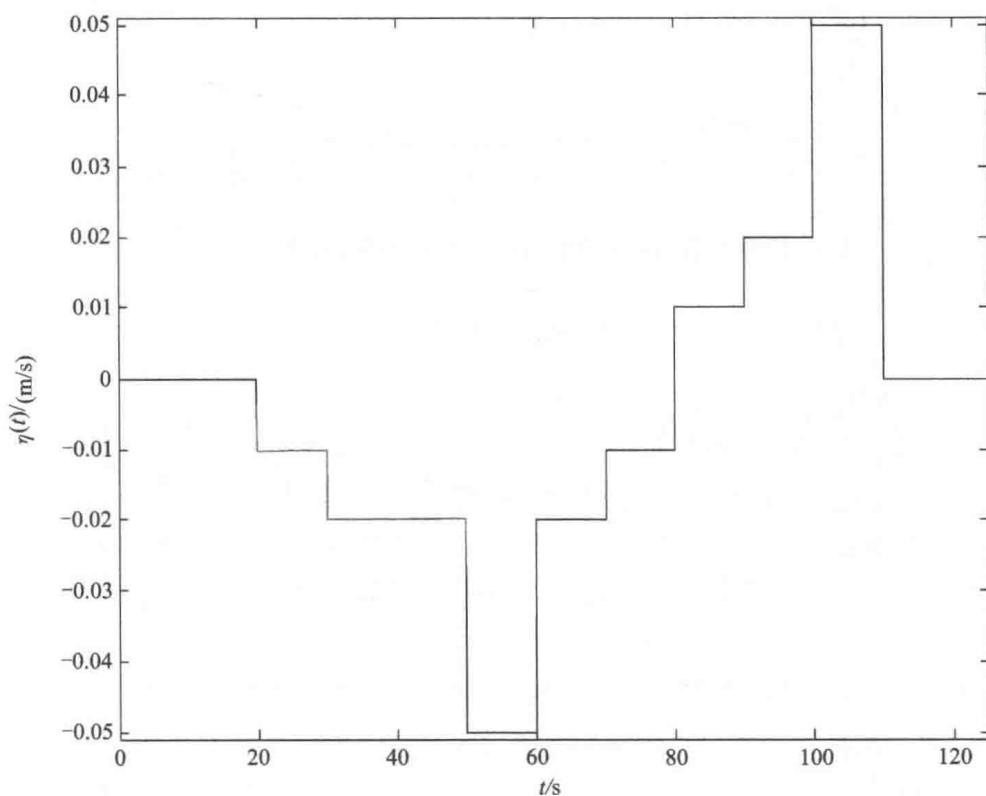
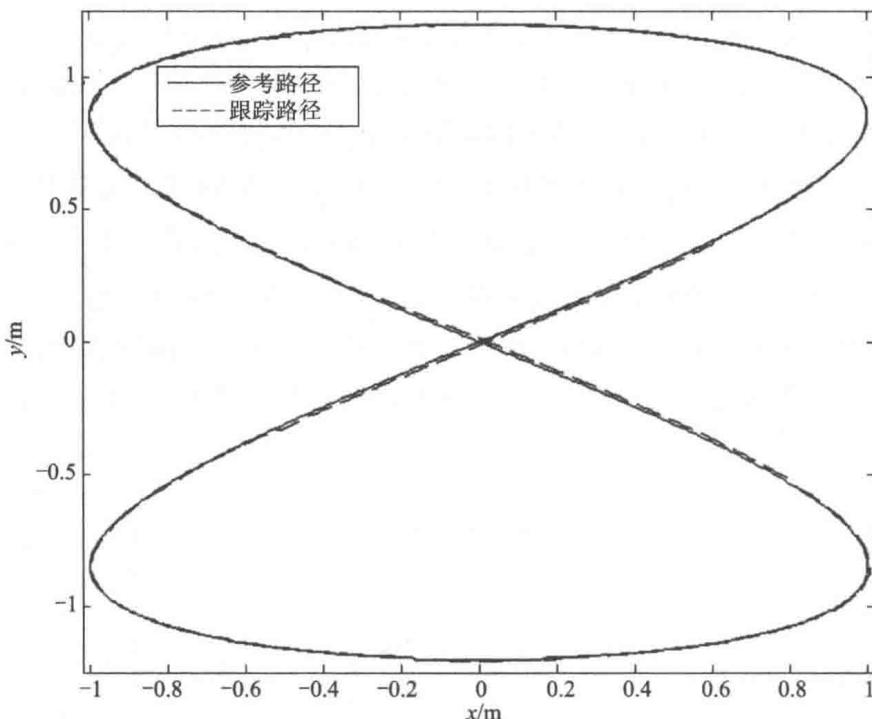
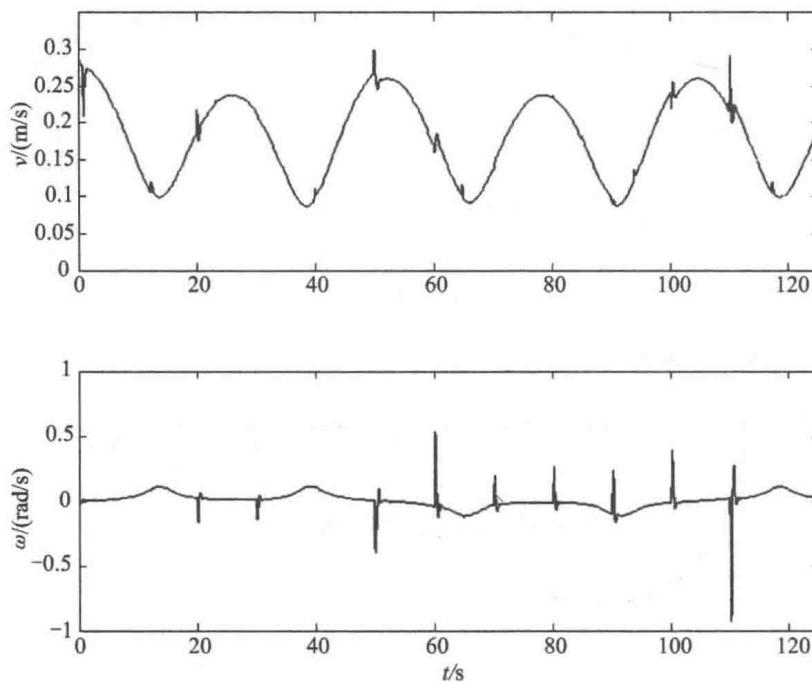


图 4-13 仿真中所采用的 $\eta(t)$

图 4-14 模型算法控制方法的路径跟踪控制仿真结果($\eta(t) \neq 0$)图 4-15 系统控制信号($\eta(t) \neq 0$)

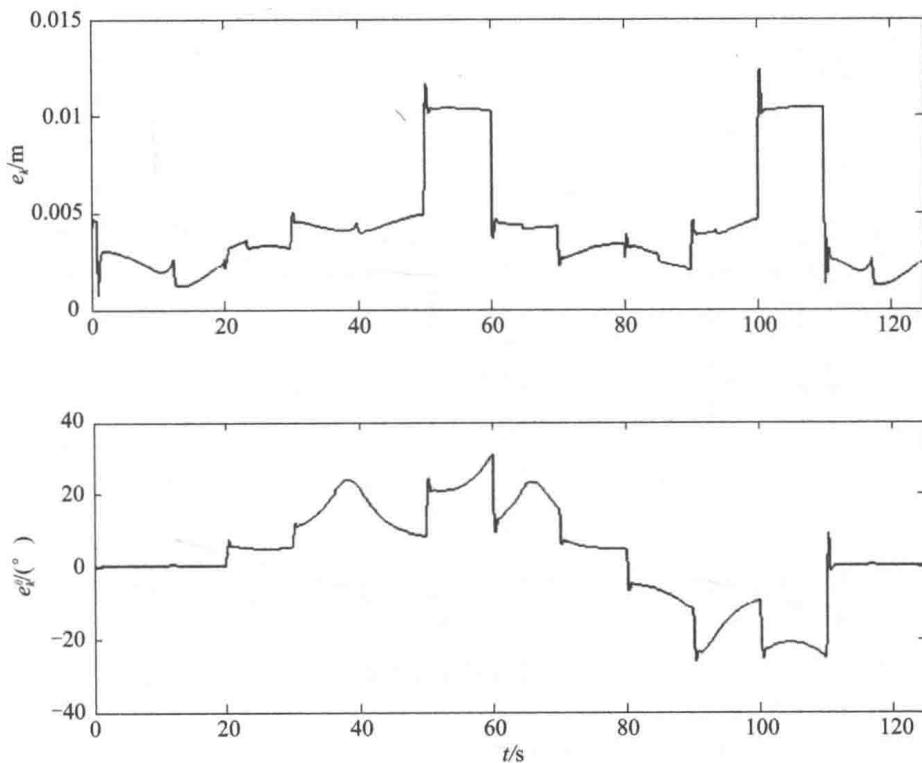


图 4-16 控制位置误差 $e_k = \sqrt{(e_k^x)^2 + (e_k^y)^2}$ 和方向角度误差 e_k^θ ($\eta(t) \neq 0$)

在进行了基于非线性模型算法控制方法设计的移动机器人运动学路径跟踪控制系统仿真后,再通过实验来进一步验证。实验所使用的机器人为我们实验室的“DU ROBO”移动机器人。该移动机器人装有自动导航系统。“DU ROBO”移动机器人拥有两个由直流电动机(两个)驱动的主动轮和一个从动轮。两个主动轮上分别配有光电编码器。移动机器人上还装有用来给出机器人当前的方向信息的电子罗盘。在实验中,系统的采样周期选为 $T=0.1$ s。实验中系统扰动 $\eta(t)$ 展示在图 4-17 中。图 4-18 展示了参考路径和采用模型算法控制方法的移动机器人路径跟踪控制结果,图 4-18 中参考路径与跟踪路径已近似重合。图 4-19 展示了移动机器人的控制位置误差 $e_k = \sqrt{(e_k^x)^2 + (e_k^y)^2}$ 和方向角度误差 e_k^θ 。

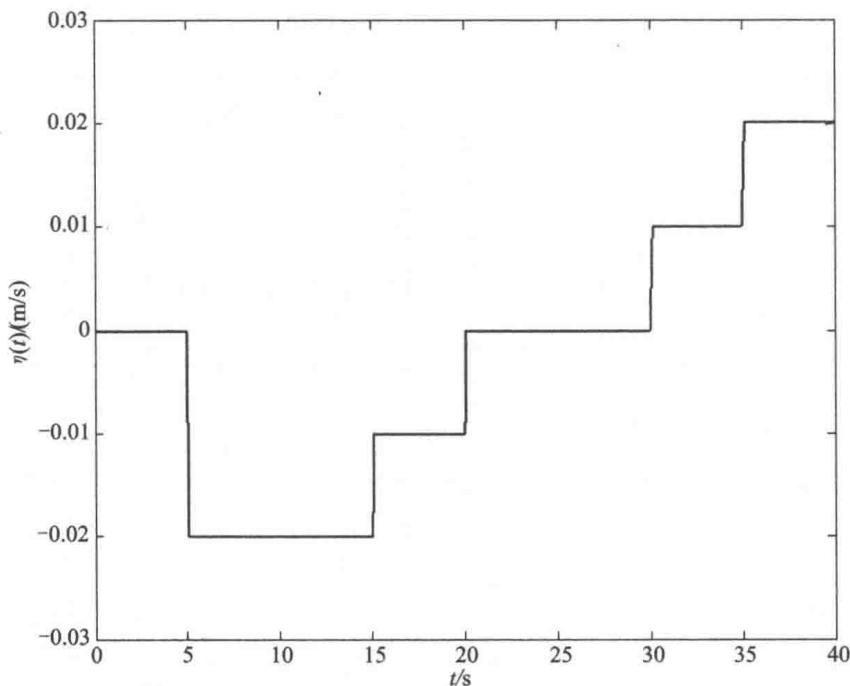
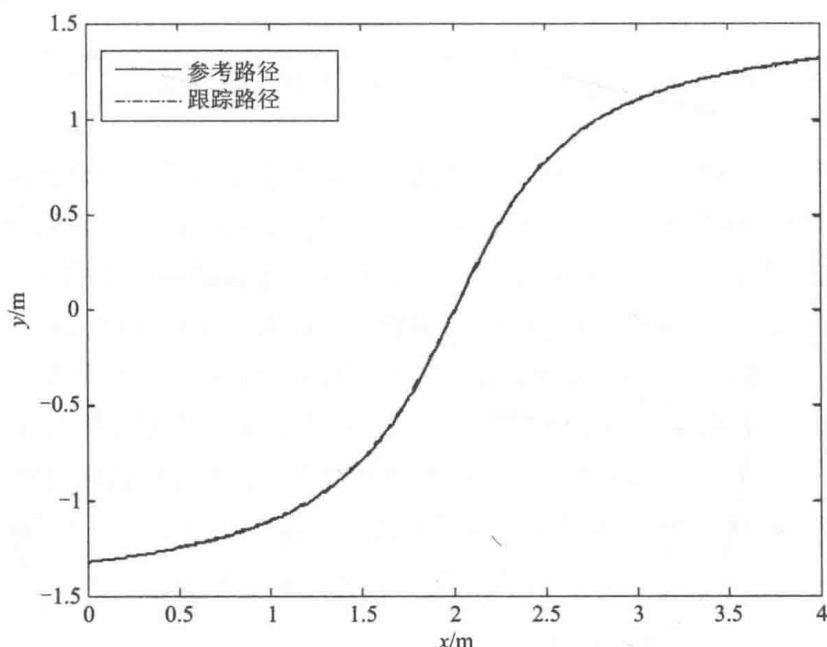
图 4-17 实验中所采用的 $\eta(t)$ 

图 4-18 模型算法控制方法的路径跟踪控制实验结果

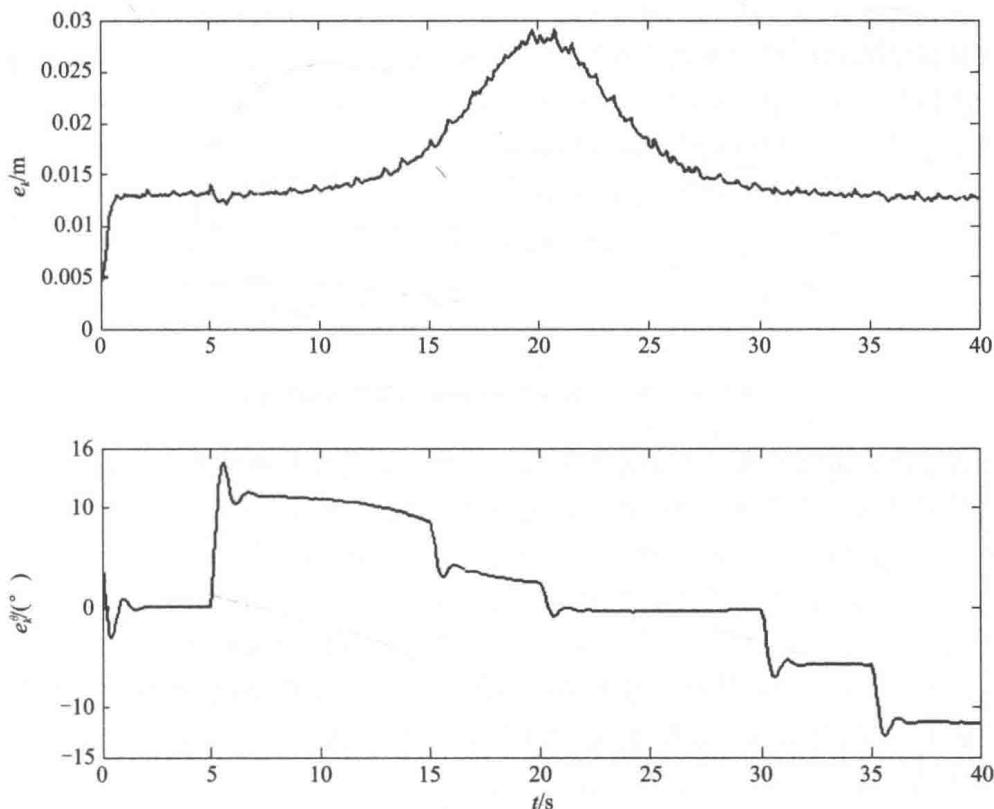


图 4-19 实验中控制位置误差 $e_k = \sqrt{(e_k^x)^2 + (e_k^y)^2}$ 和方向角度误差 e_k^θ

基于移动机器人运动学模型设计的路径跟踪控制系统,在一般情况下可以提供较理想的控制效果,但是由于运动学模型仅考虑移动机器人的位置和速度信息,而没有考虑到机器人的加速度信息和转动惯量,当机器人的载重较大或者速度较大时,仅考虑移动机器人的运动学模型将得不到很好的控制效果。这个时候应该考虑移动机器人的动力学模型。考虑式(4-49)所示的移动机器人的动力学模型。为叙述简单,式(4-49)可以简写成如式(4-80)所示形式。

$$\dot{\mathbf{X}}(t) = F[\mathbf{X}(t)] + G[\mathbf{X}(t)]\mathbf{U}(t-\mathbf{D}) \quad (4-80)$$

其中, $\mathbf{X} = [x \ y \ \theta \ v \ w]^T$, $\mathbf{U} = [u_1 \ u_2]^T$, $\mathbf{D} = [d_1 \ d_2]^T$ 是系统输入的延时时间。

采用如前所述的非线性系统离散方法可以得到轮式移动机器人的动力学模型的离散形式,如式(4-81)所示。

$$\mathbf{X}(k+1) = \Phi[\mathbf{X}(k), \mathbf{U}(k)] \quad (4-81)$$

如前所示,可以得到

$$\mathbf{U} = \Psi_0(\mathbf{X}) \quad (4-82)$$

进行模型算法控制方法的参考路径设计,最终可以得到系统的控制输入。图 4-20 展示了移动机器人路径跟踪控制系统的框图。

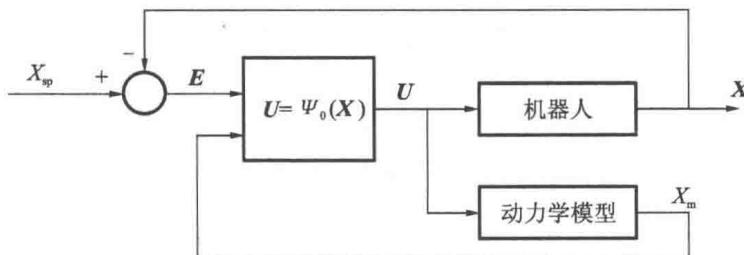


图 4-20 移动机器人路径跟踪控制系统的框图

通过仿真来验证基于非线性模型算法控制方法设计的移动机器人动力学路径跟踪控制系统。在仿真中移动机器人的物理参数为: $r = 0.15 \text{ m}$, $l = 0.5 \text{ m}$, $c = 0.25 \text{ m}$, $p = 0.15 \text{ m}$, $m_c = \hat{m}_c + \Delta m_c$, $m_w = \hat{m}_w + \Delta m_w$, $I = \hat{I} + \Delta I$, $I_w = \hat{I}_w + \Delta I_w$ 。其中, $\hat{m}_c = 5 \text{ kg}$, $\hat{m}_w = 0.05 \text{ kg}$, $\hat{I} = 0.16 \text{ kg} \cdot \text{m}$, $\hat{I}_w = 0.000016 \text{ kg} \cdot \text{m}$ 。 $|\Delta m_c| \leq \varphi_{mc}$, $|\Delta m_w| \leq \varphi_{mw}$, $|\Delta I| \leq \varphi_I$ 和 $|\Delta I_w| \leq \varphi_{Iw}$ 是系统参数的未知分量值, 其中 φ_{mc} 、 φ_{mw} 、 φ_I 和 φ_{Iw} 是已知的常量。考虑移动机器人的动力学模型, 在仿真中设置模型参数最大变化为 20%, 也就是说 $|\Delta \beta_1| \leq 0.049$, $|\Delta \beta_2| \leq 1.56$, $|\Delta \alpha_1| \leq 0.26$, $|\Delta \alpha_2| \leq 2.08$ 。

为了验证基于非线性模型算法控制方法设计的移动机器人动力学路径跟踪控制系统的有效性, 在仿真中控制移动机器人跟踪一圆形的路径, 并选取了多种的可调滤波参数 α 的值。在仿真中还在 x 和 y 上加入了标准方差为 0.005 m 的正态分布的随机干扰信号。仿真中使用的参考路径为: $x(t) = 1.5 \cos(0.1t)$, $y(t) = 1.5 \sin(0.1t)$ 。该路径由位置及其相应位置处的速度信息构成。也就是说, 当移动机器人跟踪所给定的参考路径时, 其速度信息也是预先设定好的。在这种情况下, 设置移动机器人的速度约束为 $-1 \text{ m/s} \leq v \leq 1 \text{ m/s}$ 。在仿真中模型参考控制算法的可调滤波参数分别选为 0.99、0.9、0.8 和 0.7。移动机器人的起始出发位置坐标为 $[0.5, 0, 0, 0, 0]^T$ 。基于所选取如上所述的四种模型参考控制算法的可调滤波参数的值得到的机器人路径跟踪控制结果分别展示在图 4-21、图 4-22、图 4-23 和图 4-24 中。从图中可以看出, 当可调滤波参数 $\alpha = 0.99$ 时, 机器人无法跟踪所设定的参考路径, 这是由于所选的可调参数的值太大, 而使得系统输入控制信号无法很好地调节系统。当逐渐减小可调参数的值时, 机器人的跟踪效果会逐渐变好, 机器人的跟踪速度也会逐渐增大。但是当可调参数的值减少到 0.8 时, 路径跟踪控制仿真结果中出现了超调现象。因此, 在这种情况下可调滤波参数选为 $\alpha = 0.9$ 时, 移动机器人路径跟踪控制的结果可以满足要求。模型参考控制算法的可调滤波参数的值地选取应该基于控制的目标和控制系统的特性。可以基于实验的

结果来进行可调滤波参数值的选取。从图 4-22 可以看出,当存在外部干扰信号的时候,模型算法控制方法仍然可以提供满意的路径跟踪控制结果。这是由于在设计模型算法控制方法时,设计了如式(4-54)所示的参考轨迹,使得模型算法控制方法可以通过调整系统输出与系统输出期望值之间的差值,来提高控制系统对外部干扰的鲁棒性。

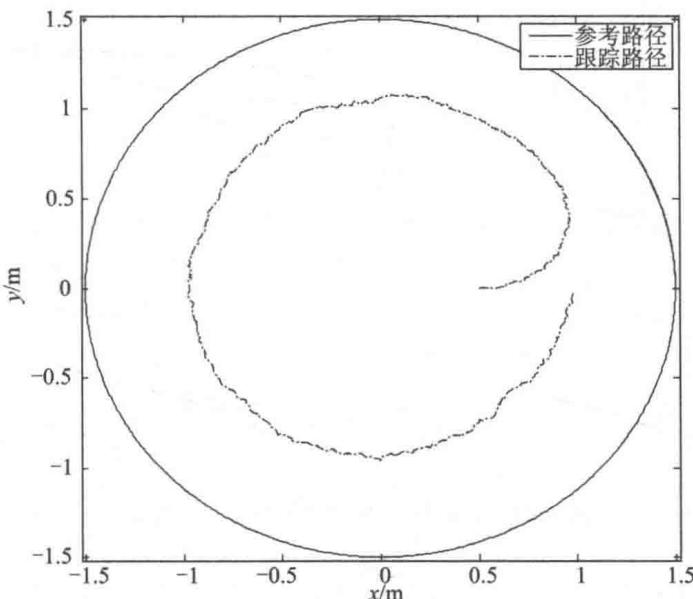


图 4-21 圆形路径跟踪控制结果 ($\alpha=0.99$)

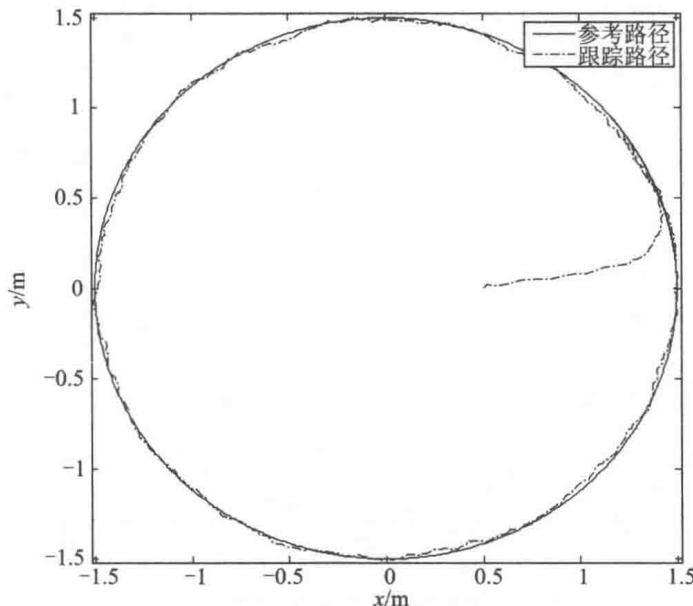


图 4-22 圆形路径跟踪控制结果 ($\alpha=0.9$)

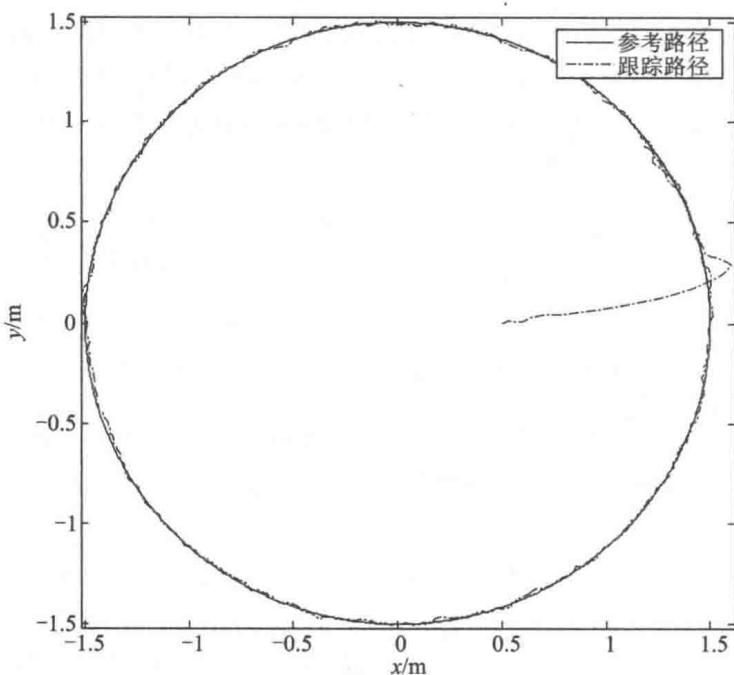


图 4-23 圆形路径跟踪控制结果($\alpha=0.8$)

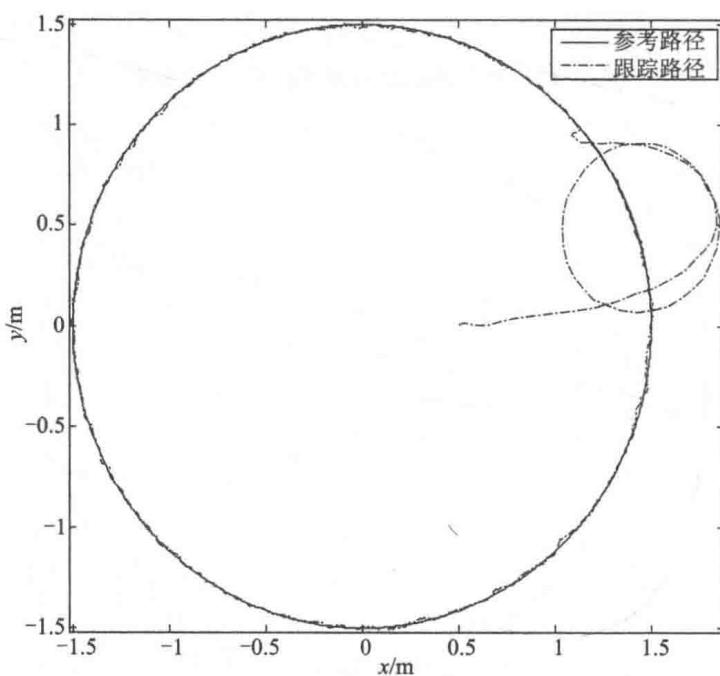


图 4-24 圆形路径跟踪控制结果($\alpha=0.7$)

在接下来的仿真中,采用一个非线性最优控制方法进行机器人路径跟踪控制仿真,将其仿真结果与基于模型算法控制方法的路径跟踪控制仿真结果做比较,以验证该方法的有效性。该非线性最优控制方法的控制思想是设计一个目标函数,然后通过求取该目标函数的最小值来得出系统的控制输入。在这种情况下,控制移动机器人跟踪一个“8”字形的参考路径。机器人的起始出发点的坐标为 $[-0.4, 0.1, 0, 0, 0]^T$ 。图4-25中展示了分别采用非线性最优控制方法和模型算法控制方法设计的机器人路径跟踪控制算法的路径跟踪控制结果。图4-26展示了采用非线性最优控制方法和模型算法控制方法设计的机器人路径跟踪控制系统的机器人的线速度和角速度。图4-27展示了控制系统的控制输入,也就是在路径跟踪控制过程中施加在机器人左右两个驱动轮上的力矩。从图4-25可以看出,利用非线性最优控制方法设计的路径跟踪控制系统无法准确地控制机器人跟踪该“8”字形的参考路径。而与其相对应的,基于模型算法控制方法设计的机器人路径跟踪控制系统可以提供很好的路径跟踪控制结果,即便仿真中机器人的起始出发点不在该参考路径上。模型算法控制方法是一种可以应用于非线性延时系统的单步预测控制器。其主要的控制思想是基于被控系统的模型,预测被控制系统未来输出与参考路径的差值,设计相应的能够反映该差值的优化标准,最终通过求取该优化标准的最小值来得到相应的系统控制输入。因此,模型算法控制方法可以提供很好的路径跟踪控制结果。通过定义模型算法控制方法的“参考轨迹”,模型算法控制方法可以对系统模型的误差和外部干扰具有一定的鲁棒性。从“8”字形路径跟踪控制仿真结果可以看出,模型算法控制方法可以提供比非线性最优控制方法更好的控制结果。

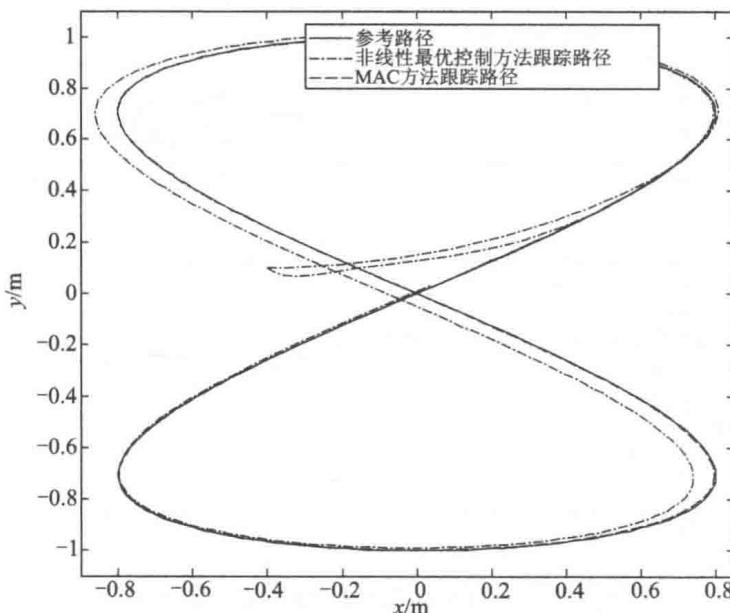


图4-25 “8”字形路径跟踪控制结果

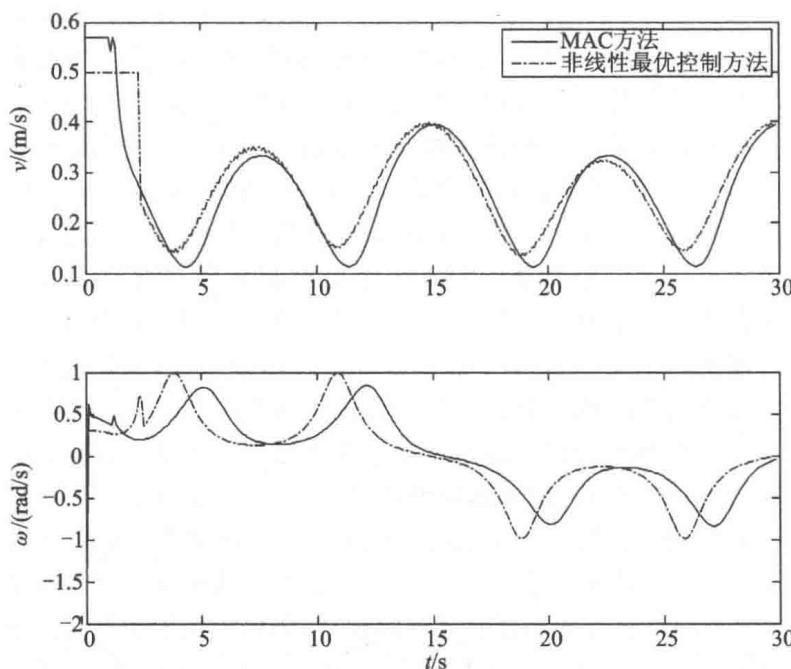


图 4-26 移动机器人的线速度和角速度

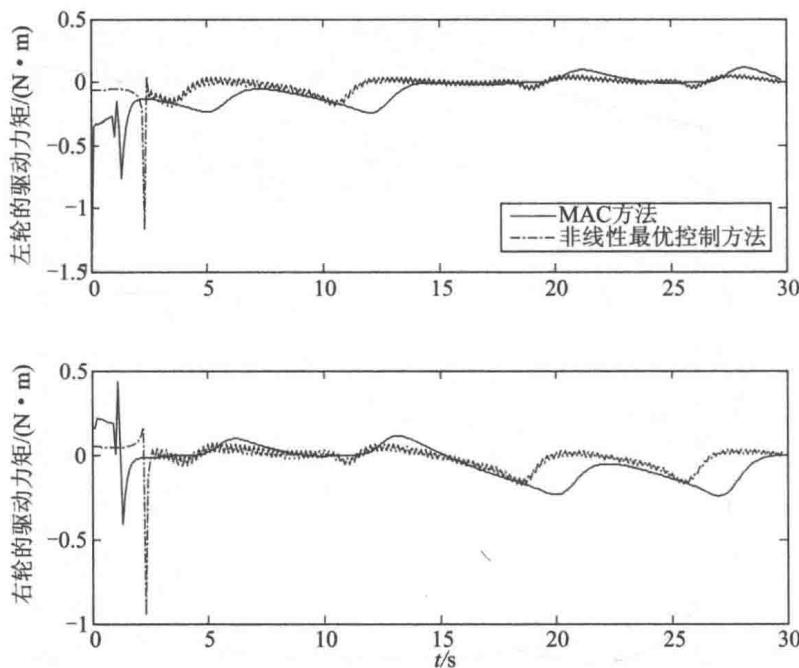


图 4-27 施加在机器人两个驱动轮上的力矩

最后,在仿真中考虑系统有延时的情况。仿真中系统的采样周期为0.1 s,系统的延时时间设置为0.84 s。仿真中的参考路径为“8”字形路径: $x(t)=0.8\sin(0.4t)$, $y(t)=1.0\sin(0.2t)$ 。机器人系统的速度约束为 $-1 \text{ m/s} \leq v \leq 1 \text{ m/s}$ 。在仿真中机器人 x 和 y 方向上分别加入了标准偏差为0.005 m的正态分布随机干扰信号。图4-28展示了路径跟踪控制的控制结果。图4-29展示了移动机器人的线速度和角速度。图4-30展示了系统的控制信号,也就是分别作用在左右两个驱动轮上的力矩。从跟踪控制结果可以看出,基于模型算法控制方法设计的移动机器人路径跟踪控制器可以在存在延时的情况下为移动机器人提供很好的路径跟踪控制。

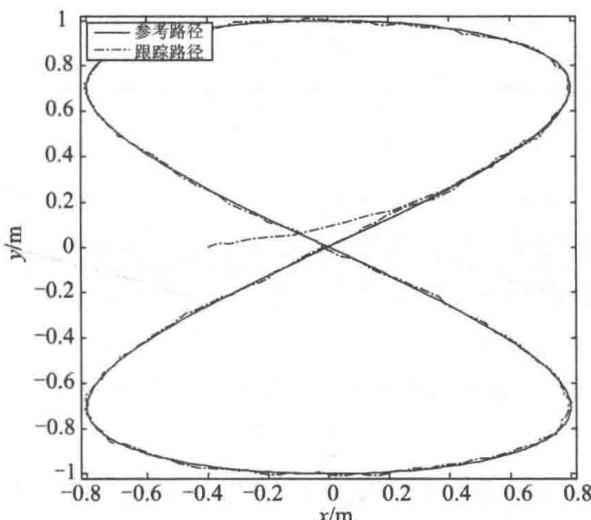


图4-28 存在延时情况下的路径跟踪控制结果

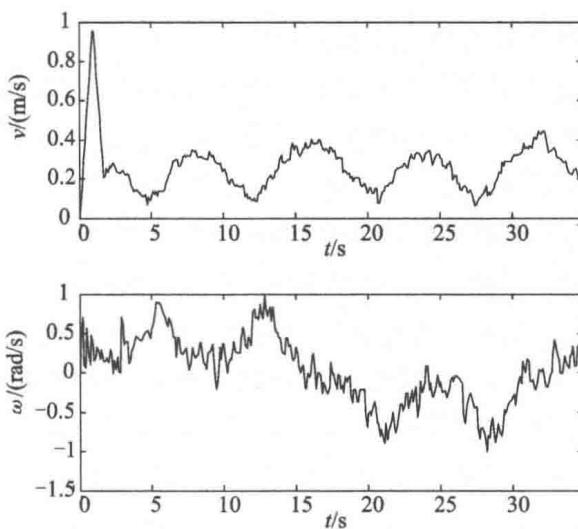


图4-29 存在延时情况下移动机器人的线速度和角速度

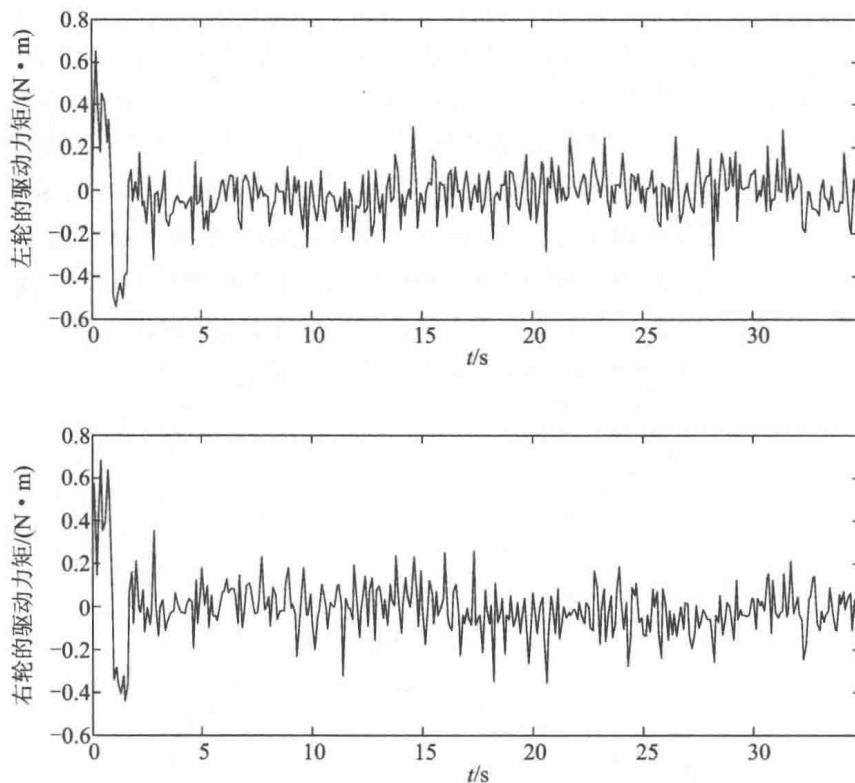


图 4-30 存在延时情况下左右两驱动轮上的驱动力矩

从图 4-28 可以看出在存在系统延时和外部干扰的情况下,基于模型算法控制方法设计的移动机器人路径跟踪控制方法可以提供满意的路径跟踪控制结果。

2) 基于神经网络的模型算法控制方法(NNMAC)

一般来说,模型算法控制算法可以为非线性系统提供最优控制结果,其控制方法是通过求取系统输出误差最小值而得到的^[105]。为了得到好的控制效果,模型算法控制方法依然需要一个相对准确的被控对象模型。然而,实际应用中大多数的被控系统具有很强的非线性特性,这使得该被控系统准确的数学模型及其相应的逆模型非常难得到。因此,如果被控系统的模型无法得到,则非线性模型算法控制方法也就无法使用。近些年来将神经网络应用到控制系统中的方法得到了广泛的研究和实际应用,其中很大一部分原因是神经网络具有很强的非线性系统模拟能力^[106]。Chen 等讨论了利用神经网络为一系列具有时变延时的不确定随机非线性系统,设计自适应输出反馈控制器的问题^[107]。他们基于系统的延时输出采用神经网络来补偿系统未知的非线性参数。Liu 等利用神经网络为不确定的非线性单输入单输出系统设计了一种自适应输出反馈控制器^[108]。Qiu 等利用反向传播

(BP)神经网络为一个柔性压电梁系统设计了一种减震控制方法^[109]。其中,神经网络用于调整 PID 控制器的参数。文献[110]中作者利用一个 BP 神经网络来调整 PID 控制器的参数,并将其应用于电厂的流体温度控制系统中。

由于很难通过传统的数学方法得到非线性延时系统的控制方法,所以我们采用两个 BP 神经网络与前述的非线性模型算法控制方法相结合的方法来设计非线性延时系统的控制方法。这两个 BP 神经网络分别训练用来模拟被控非线性系统的模型和逆模型。在得到了该非线性系统的神经网络模型和逆模型后,可以进行非线性模型算法控制方法的设计。

在设计 NNMAC 方法中使用了两个神经网络,分别命名为 NN1 和 NN2。其中神经网络 NN1 用来模拟被控非线性系统的模型,而神经网络 NN2 用来模拟该被控非线性系统的逆模型。这里所采用的神经网络为 BP 神经网络,这两个神经网络的结构采用 NNARX(autoregressive external input)结构。对于这种结构来说,神经网络的输入为过去若干个采样周期的系统输入和输出。图 4-31 展示了所采用的 BP 神经网络的结构。图 4-32 展示了 NNARX 的结构。其中参数 a 和 c 分别是过去系统输出和输入的周期数。 x 是系统的输出, u 是控制信号, x_m 是神经网络的输出, dT 是非线性系统的延时时间,其中 T 是系统的采样周期。在建立神经网络的过程中需要解决的一个重要问题是如何选择神经网络 NN1 和 NN2 的输入神经元的数量。如果神经网络输入神经元的数量选取得过小的话,则该神经网络会产生不充分拟合的问题。在这种情况下神经网络将不能很好地模拟该非线性系统。另一方面,如果神经网络神经元的数目选取得很大,神经网络的模拟准确性会得到相应的提高。但是神经网络的训练过程将变得非常的复杂。过大的神经网络神经元的数目还会导致过度拟合的问题。在这种情况下连系统的噪声也被模拟进去了,这反而会使得神经网络模型变坏。为了得到合适的神经网络神经元的数目,需要选取不同数目的神经网络神经元进行大量的训练和测试。

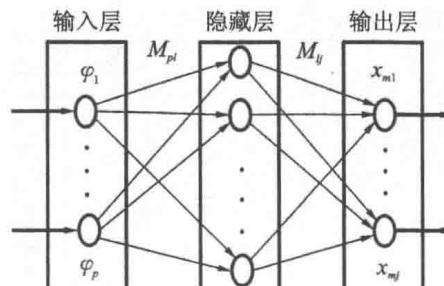


图 4-31 BP 神经网络的结构

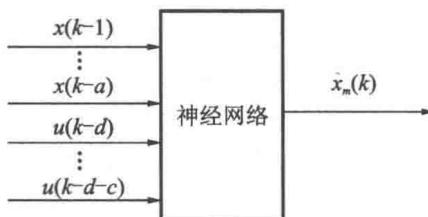
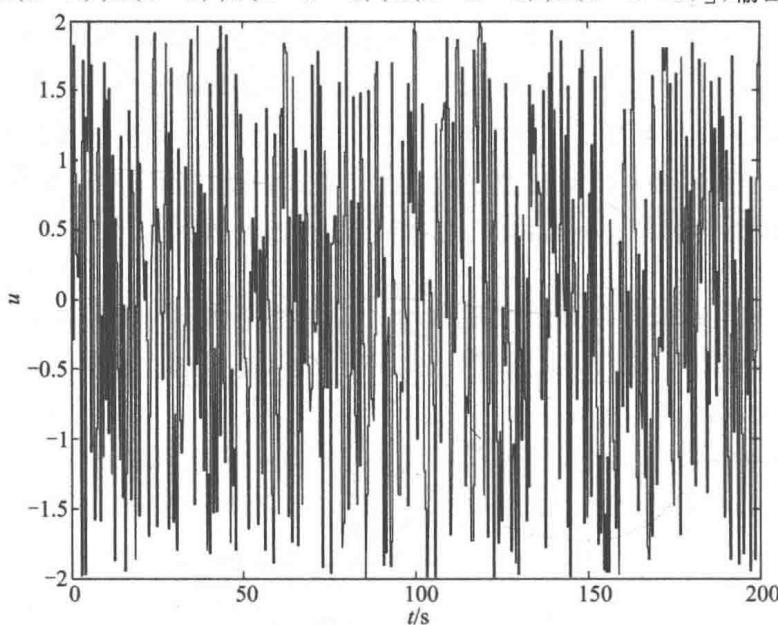


图 4-32 NNARX 的结构

采用离线训练的方式训练神经网络 NN1。首先应该做大量的系统实验来获得相应的训练和测试数据。在准备实验时必须非常细致地选取系统的输入信号 u ，使得该实验用的输入信号能够覆盖控制系统所有的工作状况。举个例子：在后面所述内容的仿真用系统 1 中，被控系统的输入信号 u 采用公式(4-83)进行设计。该输入信号展示在图 4-33 中。通过大量的实验可以得到用于训练和测试神经网络的一系列的系统输入信号 u 和相应的系统输出 x 。所得到的数据中的一半用来进行神经网络的训练，而剩下的另一半数据用来进行测试训练好的神经网络。对于后面所述的仿真用系统 1 来说，起初我们选择了较少数目的神经网络输入层神经元进行训练和测试。在发现无法得到满意的训练结果后，逐渐增大神经网络输入层神经元的数目然后查看训练和测试的效果。经过多次测试后发现，当 $a=3$ 和 $c=3$ 时可以得到满意的训练和测试结果。具体来说，神经网络 NN1 在输入层有 6 个神经元，隐藏层有 6 个神经元，输出层有 1 个神经元。隐藏层和输出层的激发函数分别为 $f_h(x) = \tanh(x)$ 和 $f_o(x) = x$ 。神经网络 NN1 的输入为 $[x(k-1), x(k-2), x(k-3), u(k-d-1), u(k-d-2), u(k-d-3)]$ ，输出为 $x_m(k)$ 。

图 4-33 用于做实验得到训练神经网络数据的系统输入 u

用于调整神经网络 NN1 权值的误差信号为被控系统的输出与神经网络 NN1 的输出的差。图 4-34 展示了训练神经网络 NN1 的结构。

$$u(m) = \text{ran} \left[\text{int} \left(\frac{m-1}{4} \right) + 1 \right], \quad m=1,2,\dots \quad (4-83)$$

其中, $\text{ran}(\quad)$ 是一均匀分布的随机数, $\text{int}[\quad]$ 表示取方括号内容整数部分。

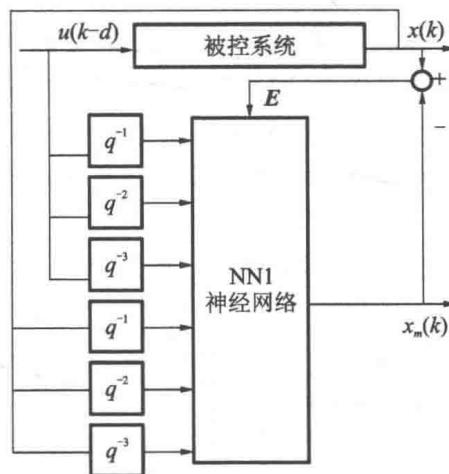


图 4-34 训练神经网络 NN1 的结构

同样采用离线训练的方式训练神经网络 NN2。确定神经网络 NN2 的输入层神经元个数的过程与确定神经网络 NN1 输入层神经元个数的过程相似。神经网络 NN2 有一个输入层,一个隐藏层和一个输出层。隐藏层和输出层的函数分别为 $f_h(x) = \tanh(x)$ 和 $f_o(x) = x$ 。选取被控系统的输出和以前采样时刻的输入作为神经网络 NN2 的输入。神经网络 NN2 的输出 $u'(k)$ 为当前采样时刻系统的输入信号的预测值。用于调整神经网络 NN2 权值的误差信号为控制系统的输入 $u(k)$ 和神经网络 NN2 的输出 $u'(k)$ 的差值。对于非线性延时系统来说,神经网络 NN2 的训练要比神经网络 NN1 的训练复杂一些。在训练神经网络 NN2 中处理系统延时的方法如下^[111]。

设系统可以用式(4-84)来表示。

$$x(k+d) = g[x(k+d-1), \dots, x(k+d-a), u(k), \dots, u(k-c)] \quad (4-84)$$

则系统输入 u 可以用式(4-85)来表示。

$$u(k) = g'^{-1}[x(k+d), x(k+d-1), \dots, x(k+d-a), u(k-1), \dots, u(k-c)] \quad (4-85)$$

考虑如前所述的非线性模型算法控制方法, $x(k+d)$ 用 $b[x_m(k), e(k)]$ 来代替, 则式(4-85)中有 $d-1$ 个未知量($[x(k+1), \dots, x(k+d-1)]$)。在如下的仿真中选取系统的延时为 $d=2$ 。在这种情况下, 式(4-85)有一个未知量, $x(k+1)$ 。

该未知量可以用式(4-86)来表示。

$$x(k+1) = g[x(k), \dots, x(k+1-a), u(k-1), \dots, u(k-c-1)] \quad (4-86)$$

考虑式(4-85)和式(4-86), $u(k)$ 可以用式(4-87)来表示。

$$u(k) = g'^{-1}[x(k+2), x(k), \dots, x(k+2-a), \dots, x(k+1-a), u(k-1), \dots, u(k-c), u(k-c-1)] \quad (4-87)$$

利用图 4-33 所示的被控系统的输入数据和其相应的系统输出数据作为神经网络 NN2 的训练和测试用数据。具体来说,对于下述仿真中的系统 1 来说,所构建神经网络的输入层有 8 个神经元,隐藏层有 8 个神经元,输出层有 1 个神经元。神经网络 NN2 的输入和输出分别为

$$\text{输入} \quad [x(k+2), x(k), x(k-1), x(k-2), x(k-3), u(k-1), u(k-2), u(k-3)]$$

$$\text{输出} \quad u'(k)$$

图 4-35 展示了训练神经网络 NN2 的结构。

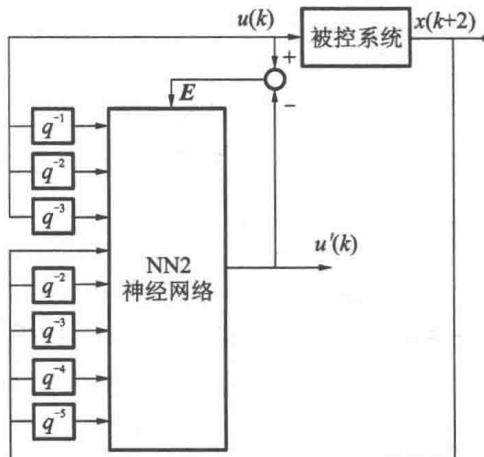


图 4-35 训练神经网络 NN2 的结构

在仿真中采用 Levenberg-Marquardt 方法来训练神经网络 NN1 和 NN2。对于模型算法控制方法来说,可以采用式(4-69)来计算系统的控制信号。在式(4-69)中参数 Ψ_0 可以用式(4-64)求得, $x_m(k)$ 可以用式(4-57)求得。类似于模型算法控制方法,神经网络模型算法控制方法利用两个神经网络 NN1 和 NN2 来提供控制信号。在这种情况下神经网络 NN1 用来模拟被控系统的模型,其输出为 $x_m(k)$; 神经网络 NN2 用来模拟被控系统的逆模型。训练好并通过测试的两个神经网络 NN1 和 NN2 可以用来构建神经网络模型算法控制方法。图 4-36 展示了神经网络模型算法控制方法框图。其中 $b[x_m(k), e(k)] = \alpha h^{-1}[x_m(k)] + (1-\alpha)\{h[x_m(k)] + e(k)\}$ 用于计算 $b[x_m(k), e(k)]$ 的值。类似于模型算法控制方法,神经网络 NN1 提供被控系统状态参量的预测值 $x_m(k)$ 。同时考虑式(4-64)和式(4-69),通过利用 $b[x_m(k), e(k)]$ 和神经网络 NN2,使神经网络 NN2 起到与式(4-69)相同的作用,为

被控非线性延时系统提供控制信号。

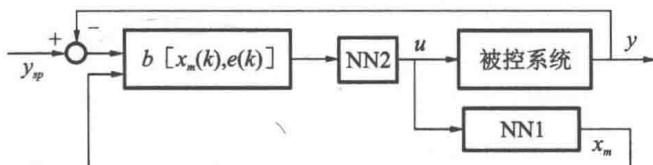


图 4-36 神经网络模型算法控制系统框图

下面利用仿真来对神经网络模型算法控制方法进行验证。在仿真中考虑采用非线性延时系统。延时的控制系统具有复杂的特性,这是由于其无限维数的特点。延时因素或多或少会对控制系统的特性产生不良的影响。在众多的实际工业系统中由于有延时现象的存在,导致系统不稳定,产生振荡。在控制系统的设计中,特别是非线性控制系统,如果不考虑延时的影响,将会产生无法预期的不良影响。

系统 1:采用式(4-77)所示的完全混合连续搅拌釜反应器系统。首先,采用如前所述的结构和方法,训练并测试神经网络 NN1 和 NN2 来预测被控系统的模型和逆模型。系统的采样周期选为 $T=0.1$ s。图 4-37 和图 4-38 分别展示了神经网络 NN1 和 NN2 的训练和测试的误差。神经网络 NN1 训练和测试的均方差分别为 1.0567×10^{-8} 和 8.5423×10^{-9} 。神经网络 NN2 训练和测试的均方差分别为 3.0316×10^{-6} 和 2.6475×10^{-6} 。训练好并测试通过的神经网络 NN1 和 NN2 用来构建神经网络模型算法控制系统。

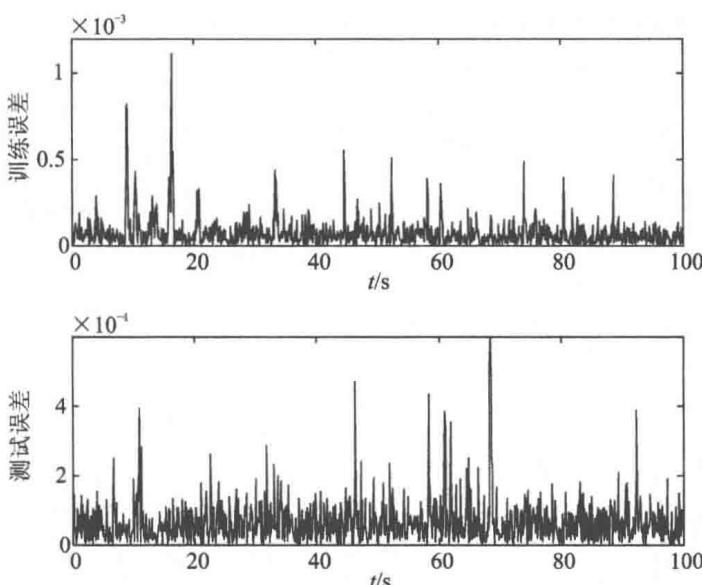


图 4-37 神经网络 NN1 的训练和测试误差

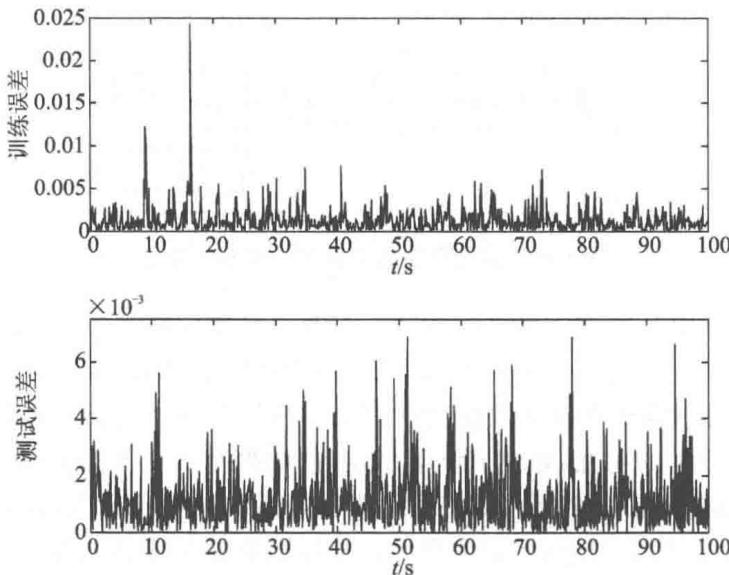
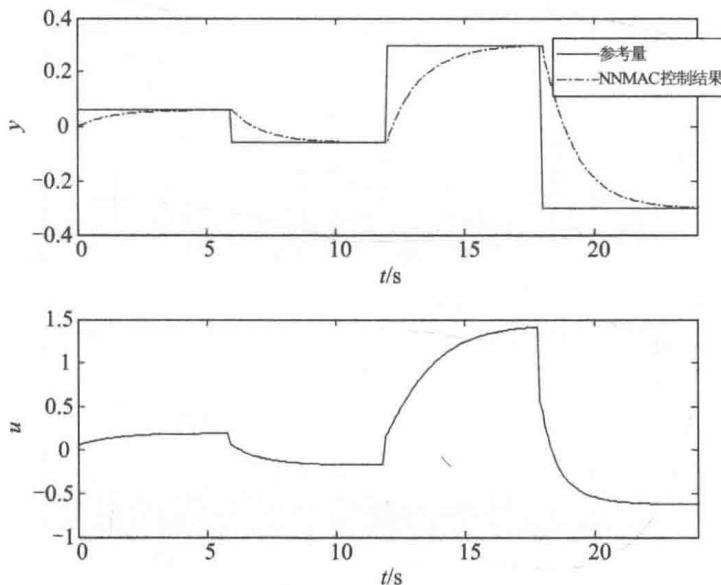


图 4-38 神经网络 NN2 的训练和测试误差

最开始,选取系统的初始值为 $x(0)=0.02$,系统的采样周期为 $T=0.1$ s,系统的延时时间为 $D=0.1$ s。利用一个非线性逆控制器与神经网络模型算法控制方法进行比较。图 4-41 展示了两个控制方法的比较结果。

首先,设置 $\alpha=0.9$ 。图 4-39 展示了控制系统的输出和控制信号。

图 4-39 控制系统的输出和控制信号($D=0.1$ s, $\alpha=0.9$)

从图 4-39 可以看出,由于 α 选取的比较大从而导致了被控系统的输出跟踪参考信号的速度很慢。很明显 $\alpha=0.9$ 不是一个好的选择。然后仿真中逐渐减小 α 的值。图 4-40 展示了当 $\alpha=0.7$ 时控制系统的输出和控制信号。

然后,选取 $\alpha=0.5$ 。在这种情况下采用一个非线性逆控制器与神经网络模型算法控制方法进行仿真结果的比较。图 4-41 展示了当 $\alpha=0.5$ 时两种控制方法的控制结果。

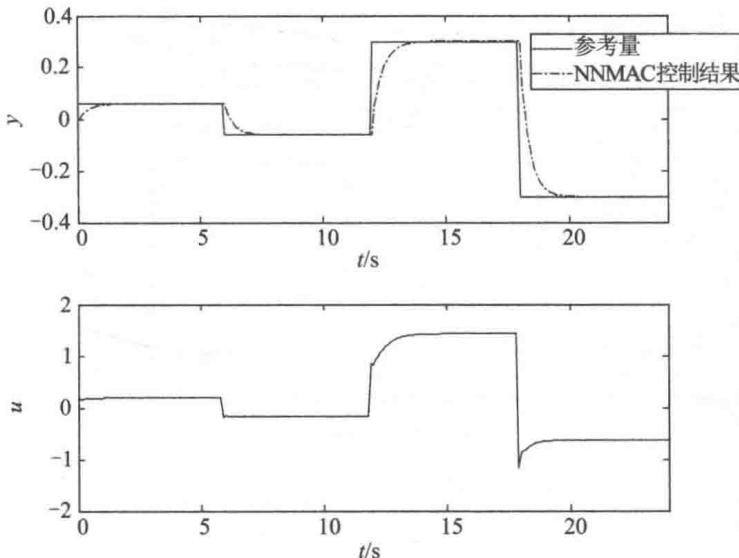


图 4-40 控制系统的输出和控制信号($D=0.1$ s, $\alpha=0.7$)

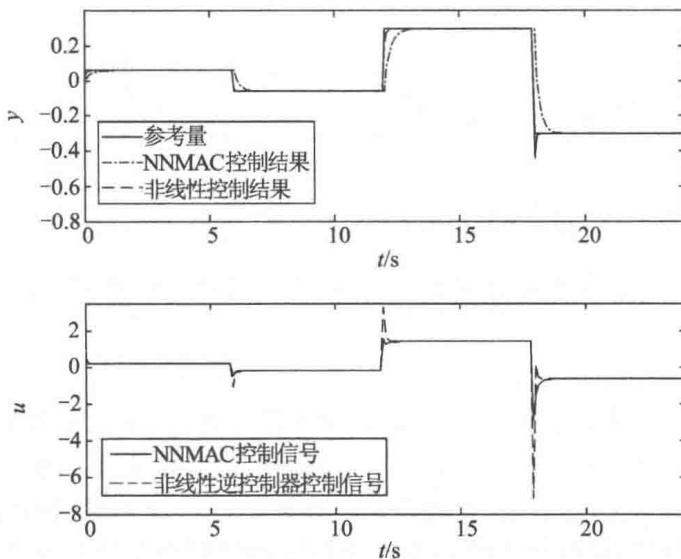


图 4-41 两种控制器作用下的控制系统的输出和控制信号($D=0.1$ s, $\alpha=0.5$)

从图 4-39 和图 4-40 可以看出当 α 的值减小时被控系统跟踪参考信号的速度会增加, 同时系统的控制信号也相应变大。 α 的值可以根据控制要求和被控系统的特性进行设置。在仿真用系统 1 的情况下, $\alpha=0.5$ 可以提供满意的控制结果。图 4-41 展示了分别采用一非线性逆控制器和神经网络模型算法控制方法设计的控制器的控制结果。从图 4-41 可以看出基于神经网络模型算法控制方法设计的控制器的控制效果要明显好于该非线性逆控制器的控制效果。

最后仿真用的参数设置为 $x(0)=0.02$, $T=0.1$ s, $\alpha=0.5$, $D=0.12$ s。图 4-42 展示了这种情况下控制系统的输出和控制信号。

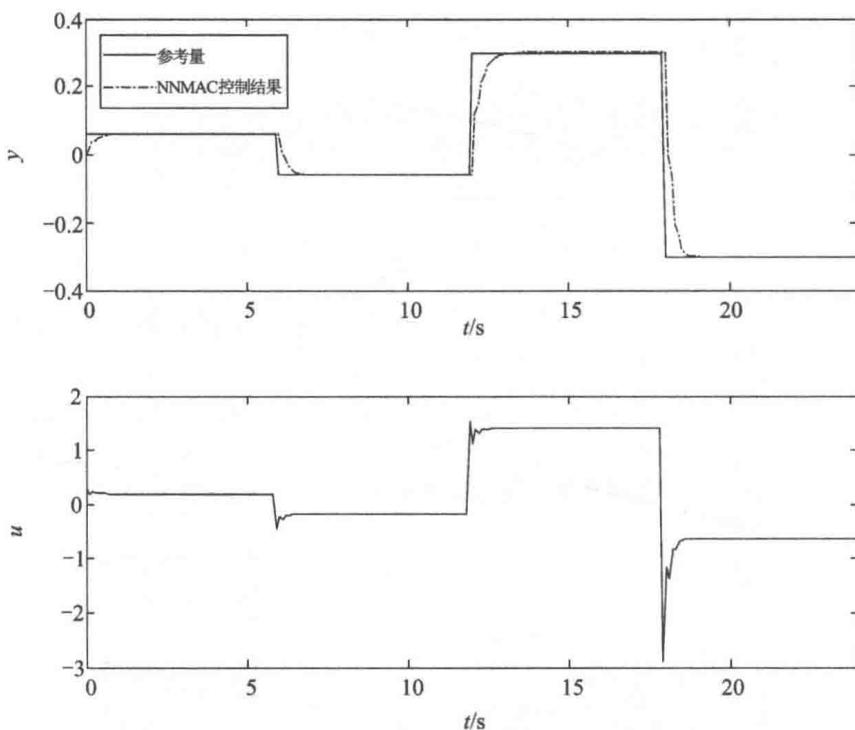


图 4-42 控制系统的输出和控制信号 ($D=0.12$ s, $\alpha=0.5$)

系统 2: 控制一个一关节机械臂^[112]。该系统的动力学模型可以用式(4-88)来表示。

$$ml^2 \ddot{q} + b \dot{q} + mlg_v \cos(q) = u \quad (4-88)$$

其中, 机械臂的长度为 l , 质量为 m , q 是机械臂的角度, g_v 为与重量有关的参数, 初始值设置为 $q(0)=0.1$, $\dot{q}(0)=0$ 。式(4-88)所示系统模型的相关参数设置为 $m=1.2$, $l=1.5$, $g_v=1$, $b=1.1$ 。系统的采样周期为 $T=0.2$ s, 延时时间为 $D=0.2$ s。在该仿真中分别采用模型算法控制方法和神经网络模型算法控制方法进行控制。在系统 2 的仿真中设置系统受到标准差为 0.06 rad 的正态分布随机外部扰动的影响。图 4-43 展示了控制系统的输出和控制信号。

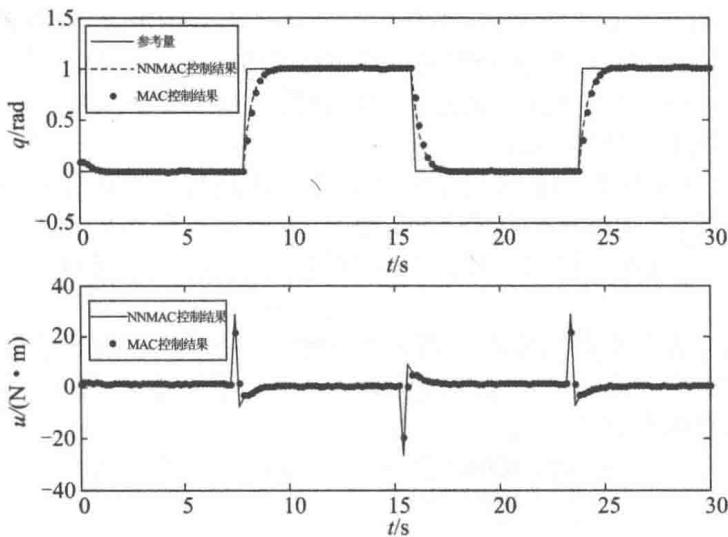


图 4-43 两种控制方法的控制系统输出和控制信号

系统 3: 考虑一个倒立摆系统^[113]。式(4-89)表示了该系统的动力学模型。

$$\ddot{\theta} = \frac{g \sin \theta - \cos \theta \{ (mL/(M+m)) \dot{\theta}^2 \sin \theta - [1/(M+m)]u \}}{(4/3)L - [mL/(M+m)] \cos^2 \theta} \quad (4-89)$$

其中, θ 是倒立摆与中垂线间的夹角, u 是系统的输入力, $g = 9.8 \text{ m/s}^2$ 是重力加速度, $M = 1.0 \text{ kg}$ 是小车的质量, $m = 0.5 \text{ kg}$ 是倒立摆的质量, $l = 0.5 \text{ m}$ 是倒立摆长度的一半。系统的采样周期为 $T = 0.2 \text{ s}$, 延时时间为 $D = 0.2 \text{ s}$ 。在仿真中设该系统受到标准偏差为 0.58 rad 的正态分布随机外部扰动的影响。分别采用模型算法控制方法和神经网络模型算法控制方法进行仿真。图 4-44 展示了控制系统的输出和相应的控制信号。

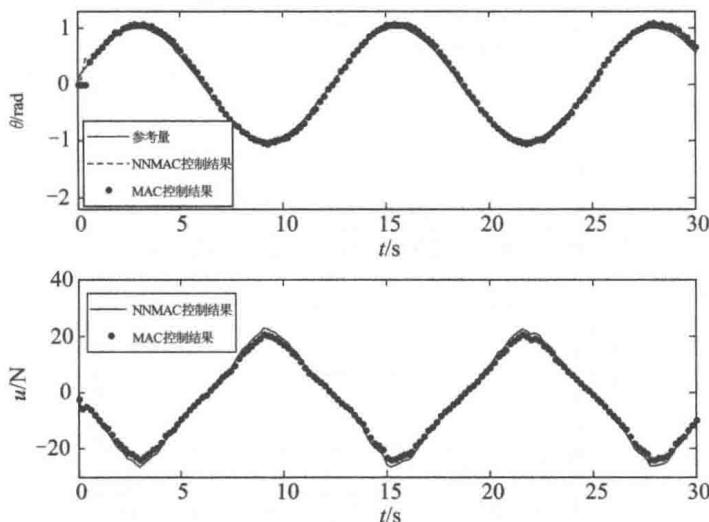


图 4-44 两种控制方法的控制系统输出和控制信号

从系统 2 和系统 3 可以看出,当存在外部扰动影响的情况下,神经网络模型算法控制方法可以提供比模型算法控制方法更好的控制效果。特别是当系统的模型不准确时,神经网络模型算法控制方法的优势将更加的明显,这是由于神经网络具有优秀的非线性系统模拟的能力。

最后,基于神经网络模型算法控制方法进行移动机器人路径跟踪控制的仿真验证。如前所述,先训练神经网络 NN1 和 NN2 来模拟移动机器人的模型和逆模型,接下来基于训练好并通过测试的神经网络 NN1 和 NN2 进行路径跟踪控制系统的设计。

首先仿真中控制机器人跟踪一圆形参考路径。图 4-45 展示了路径跟踪控制的结果,图 4-45 中参考路径与跟踪路径的一部分已近似重合,图 4-46 展示了移动机器人的线速度和角速度。

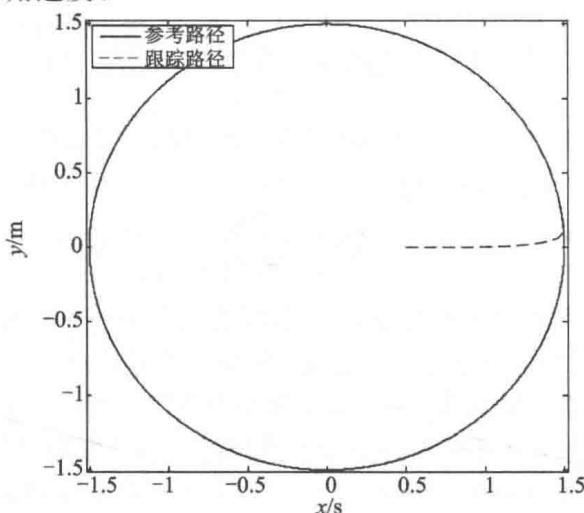


图 4-45 圆形路径的跟踪控制结果

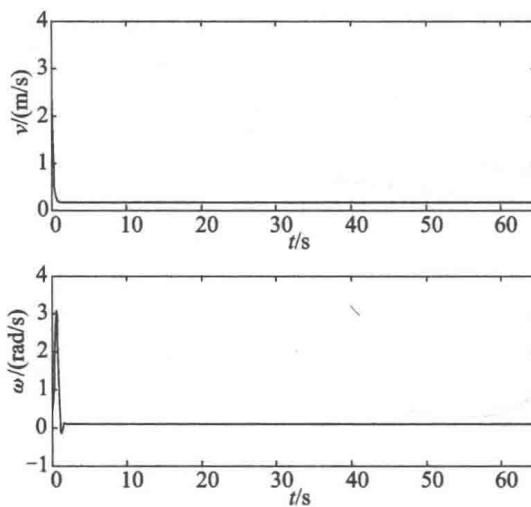


图 4-46 跟踪圆形路径情况下移动机器人的线速度和角速度

然后控制移动机器人跟踪一条“8”字形参考路径。图4-47展示了“8”字形参考路径跟踪控制的结果,图4-47中参考路径与跟踪路径的一部分已近似重合,图4-48展示了这种情况下移动机器人的线速度和角速度。

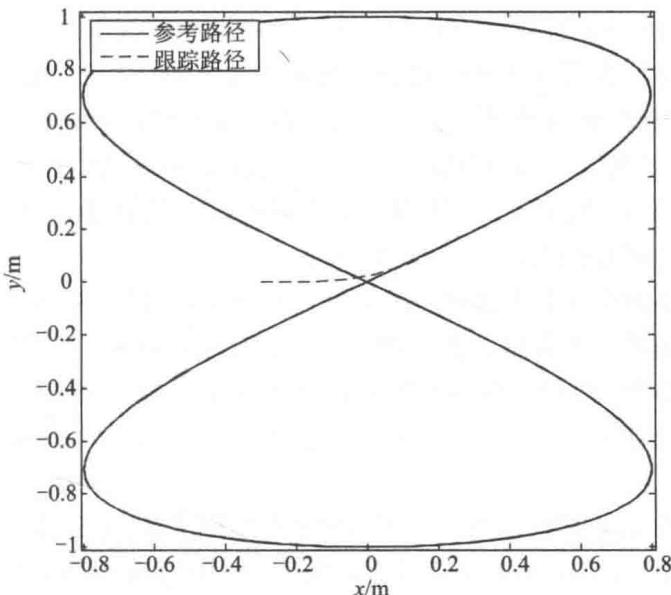


图4-47 “8”字形路径的跟踪控制结果

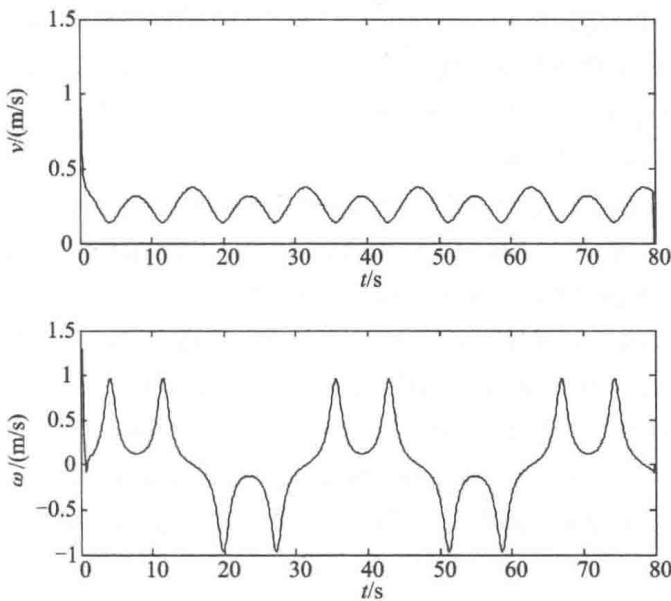


图4-48 跟踪“8”字形路径情况下移动机器人的线速度和角速度

从上述仿真结果可以看出,神经网络模型算法控制方法可以为移动机器人提供满意的路径跟踪控制结果。

参 考 文 献

- [1] 李灿. 智能移动机器人导航策略研究与实现[D]. 浙江:浙江工业大学,2013.
- [2] 吴雄英,胡国清,黄玉程,等. 自主移动机器人体系结构状况与发展研究[J]. 自动化博览,2005,22(3):21-24.
- [3] 安秋,周俊,姬长英. 基于 CAN 总线的农业轮式移动机器人[C]. 黑龙江:中国农业工程学会学术年会,2007.
- [4] 张亚利. 水下声学定位和导航技术[J]. 海洋技术,1983,(2):46-54.
- [5] 陈卫民. 美国第一代导航卫星系统——子午仪[J]. 国际太空,1979(5):1-14.
- [6] HOU K, SUN H, JIA Q, et al. An autonomous positioning and navigation system for spherical mobile robot[J]. Procedia Engineering, 2012, 29 (4): 2556-2561.
- [7] NOURELDIN A, EL-SHAFIE A, BAYOUMI M. GPS/INS integration utilizing dynamic neural networks for vehicular navigation[J]. Information Fusion, 2011, 12(1):48-57.
- [8] SEO W, HWANG S, PARK J, et al. Precise outdoor localization with a GPS-INS integration system[J]. Robotica, 2013, 281(31):371-379.
- [9] STANCIC R, GRAOVAC S. The integration of strap-down INS and GPS based on adaptive error damping[J]. Robotics & Autonomous Systems, 2010, 58(10):1117-1129.
- [10] ADUSUMILLI S, BHATT D, WANG H, et al. A low-cost INS/GPS integration methodology based on random forest regression[J]. Expert Systems with Applications, 2013, 40(11):4653-4659.
- [11] MIZUSHIMA A, ISHII K, NOGUCHI N, et al. Development of a low-cost attitude sensor for agricultural vehicles[J]. Computers and Electronics in Agriculture, 2011, 76(2):198-204.
- [12] HOU K, SUN H, JIA Q, et al. An autonomous positioning and navigation system for spherical mobile robot[J]. Procedia Engineering, 2012, 29 (4): 2556-2561.
- [13] HANZEL J, KI' ÚČIK M, DUCHOŇ F, et al. Localization of small mobile robot by low-cost GPS receiver[J]. 机械工程与自动化:英文版, 2013(8): 522-528.
- [14] NISHIDA T, INOUE S, SAGARA S. State estimation of mobile robot using

GPS signal that includes reflected and diffracted waves[J]. *Artificial Life & Robotics*, 2013, 18(3-4): 178-186.

[15] ROVIRA-MÁS F, CHATTERJEE I, SÁIZ-RUBIO V. The role of GNSS in the navigation strategies of cost-effective agricultural robots[J]. *Computers & Electronics in Agriculture*, 2015, 112: 172-183.

[16] PAZDERSKI D, DUTKIEWICZ P. Low-cost GPS Receivers in Navigation of Mobile Robots[J]. *Proceedings of Third International Workshop on Robot Motion and Control*, 2002, 11(11): 119-122.

[17] JURIŠICA L, VITKO A, DUCHOŇ F, et al. Statistical approach to GPS positioning of mobile robot[J]. *Control Engineering & Applied Informatics*, 2010, 12(2): 44-51.

[18] BAKKER T, ASSELT K V, BONTSEMA J, et al. Autonomous navigation using a robot platform in a sugar beet field[J]. *Biosystems Engineering*, 2011, 109(4): 357-368.

[19] LEE K, PARK J, KHATIB O, et al. Feedforward global/inertial sensor fusion algorithm for accurate global positioning of a mobile robot[C]//IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings. IEEE, 2005: 504-509.

[20] BORENSTEIN J, FENG L. Measurement and Correction of Systematic Odometry Errors in Mobile Robots[J]. *IEEE Transactions on Robotics & Automation*, 1997, 12(6): 869-880.

[21] ALMAGBILE A, WANG J, DING W. Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration[J]. *Cpgps*, 2010, 9(1): 33-40.

[22] HAN S, WANG J. Integrated GPS/INS navigation system with dual-rate Kalman filter[J]. *GPS Solutions*, 2012, 16(3): 389-404.

[23] CHEN X, SHEN C, ZHANG W, et al. Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages [J]. *Measurement*, 2013, 46(10): 3847-3854.

[24] STANČIĆ R, GRAOVAC S. The integration of strap-down INS and GPS based on adaptive error damping[J]. *Robotics & Autonomous Systems*, 2010, 58(10): 1117-1129.

[25] NOURELDIN A, EL-SHAFIE A, BAYOUMI M. GPS/INS integration utilizing dynamic neural networks for vehicular navigation[J]. *Information Fusion*, 2011, 12(1): 48-57.

- [26] AGHILI F, SALERNO A. Driftless 3-D attitude determination and positioning of mobile robots by integration of IMU with two RTK GPSs[J]. IEEE/ASME Transactions on Mechatronics, 2013, 18(1): 21-31.
- [27] MILLER I, CAMPBELL M. Sensitivity analysis of a tightly-coupled GPS/INS system for autonomous navigation[J]. IEEE Transactions on Aerospace & Electronic Systems, 2012, 48(2): 1115-1135.
- [28] CHO S Y, WAN S C. Robust positioning technique in low-cost DR/GPS for land navigation[J]. IEEE Transactions on Instrumentation & Measurement, 2006, 55(4): 1132-1142.
- [29] AKKAYA R. An ANN based NARX GPS/DR system for mobile robot positioning and obstacle avoidance[J]. 2013, 1(1): 6-13.
- [30] KALA R, SHUKLA A, TIWARI R. Robotic path planning in static environment using hierarchical multi-neuron heuristic search and probability based fitness[J]. Neurocomputing, 2011, 74(14-15): 2314-2335.
- [31] ZHANG Y, GONG D, ZHANG J. Robot path planning in uncertain environment using multi-objective particle swarm optimization[J]. Neurocomputing, 2013, 103(2): 172-185.
- [32] MONTIEL O, OROZ(O-ROSAS U, SEPULVEDA R. Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles[J]. Expert Systems with Applications, 2015, 42(12): 5177-5191.
- [33] RASHID A T, ALI A A, FRASCA M, et al. Path planning with obstacle avoidance based on visibility binary tree algorithm[J]. Robotics & Autonomous Systems, 2013, 61(12): 1440-1449.
- [34] FRANTISEK D, ANDREJ B, MARTIN K, et al. Path planning with modified a star algorithm for a mobile robot[J]. Procedia Engineering, 2014, 96 (96): 59-69.
- [35] AZIMIRAD V, SHORAKAEI H. Dual hierarchical genetic-optimal control: A new global optimal path planning method for robots[J]. Journal of Manufacturing Systems, 2013, 33(1), 139-148.
- [36] DAKULOVIC M, PETROVIC I. Two-way D*, algorithm for path planning and replanning [J]. Robotics & Autonomous Systems, 2011, 59 (59): 329-342.
- [37] HAWA M. Light-assisted A*, Path planning[J]. Engineering Applications of Artificial Intelligence, 2013, 26(2): 888-898.
- [38] ZHANG Y, GONG D W, ZHANG J H. Robot path planning in uncertain en-

vironment using multi-objective particle swarm optimization[J]. Neurocomputing, 2013, 103(2): 172-185.

[39] KALA R, SHUKLA A, TIWARI R. Robotic path planning in static environment using hierarchical multi-neuron heuristic search and probability based fitness[J]. Neurocomputing, 2011, 74(14-15): 2314-2335.

[40] QU H, XING K, ALEXANDER T. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots[J]. Neurocomputing, 2013, 120(10): 509-517.

[41] MO H, XU L. Research of biogeography particle swarm optimization for robot path planning[J]. Neurocomputing, 2015, 148(148): 91-99.

[42] MONTIEL O, OROZCO-ROSAS U, SEPÚLVEDA R. Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles[J]. Expert Systems with Applications, 2015, 42(12): 5177-5191.

[43] ESPELOSÍN J, ACOSTA L, ALONSO D. Path planning approach based on flock dynamics of moving particles[J]. Applied Soft Computing, 2013, 13(4): 2159-2170.

[44] HUANG C J, WANG Y W, CHEN H M, et al. Application of cellular automata and type-2 fuzzy logic to dynamic vehicle path planning[J]. Applied Soft Computing, 2014, 19(2): 333-342.

[45] CONTRERAS-CRUZ M A, AYALA-RAMIREZ V, HERNANDEZ-BEL-MONTE U H. Mobile robot path planning using artificial bee colony and evolutionary programming[J]. Applied Soft Computing, 2015, 30: 319-328.

[46] LIANG J. A path planning algorithm of mobile robot in known 3D environment[J]. Procedia Engineering, 2011, 15: 157-162.

[47] GE S S, CUI Y J. New potential functions for mobile robot path planning [J]. IEEE Transactions on Robotics & Automation, 2000, 16(5): 615-620.

[48] LI Q, ZHANG C, HAN C, et al. Path planning based on fuzzy logic algorithm for mobile robots in static environment[C]//Control and Decision Conference. IEEE, 2013: 2866-2871.

[49] QUAH, YANG S X, WILLIAMS A R, et al. Real-time robot path planning based on a modified pulse-coupled neural network model[J]. IEEE Transactions on Neural Networks, 2009, 20(11): 1724-1739.

[50] SCHYNS M. An ant colony system for responsive dynamic vehicle routing [J]. European Journal of Operational Research, 2015, 245(3): 704-718.

[51] GARCIA M A P, MONTIEL O, CASTILLO O, et al. Path planning for au-

tonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation [J]. *Applied Soft Computing*, 2009, 9 (3): 1102-1110.

[52] TAN G Z, HE H, AARON S. Ant colony system algorithm for realtime globally optimal path planning of mobile robots[J]. *自动化学报*, 2007, 33 (3): 279-285.

[53] WANG Z Q, ZHU X G, HAN Q Y. Mobile robot path planning based on parameter optimization ant colony algorithm[J]. *Procedia Engineering*, 2011, 15(1):2738-2741.

[54] ZHONG Z S. Ant colony algorithm based on path planning for mobile agent migration[J]. *Procedia Engineering*, 2011, 23:1-8.

[55] IOANNIDIS K, SIRAKOULIS G C, ANDREADIS I. Cellular ants: A method to create collision free trajectories for a cooperative robot team[J]. *Robotics & Autonomous Systems*, 2011, 59(2):113-127.

[56] ZHU Q, HU J, CAI W, et al. A new robot navigation algorithm for dynamic unknown environments based on dynamic path recomputation and an improved scout ant algorithm [J]. *Applied Soft Computing*, 2011, 11 (8): 4667-4676.

[57] SUDHOLT D, THYSSEN C. Running time analysis of ant colony optimization for shortest path problems[J]. *Journal of Discrete Algorithms*, 2012, 10 (10):165-180.

[58] CORRADINI M L, LEO T, ORLANDO G. Experimental testing of a discrete-time sliding mode controller for trajectory tracking of a wheeled mobile robot in the presence of skidding effects[J]. *Journal of Robotic Systems*, 2002, 19(4):177-188.

[59] DO K D. Bounded controllers for global path tracking control of unicycle-type mobile robots[J]. *Robotics & Autonomous Systems*, 2013, 61 (8): (2013), pp. 775-784.

[60] MOHARERI O, DHAOUADI R, RAD A B. Indirect adaptive tracking control of a nonholonomic mobile robot via neural Networks[J]. *Neurocomputing*, 2012, 88(7):54-66.

[61] HUANG J, WEN C, WANG W, et al. Adaptive Output Feedback Tracking Control of A Nonholonomic Mobile Robot[J]. *Automatica*, 2014, 50(3):821-831.

[62] MOUSTRISGP, TZAFESTAS S G. Switching fuzzy tracking control for

mobile robots under curvature constraints[J]. *Control Engineering Practice*, 2011, 19(1): 45-53.

[63] HUANG J, WEN C, WANG W, et al. Adaptive stabilization and tracking control of a nonholonomic mobile robot with input saturation and disturbance[J]. *Systems & Control Letters*, 2013, 62(3): 234-241.

[64] TZAFESTAS S G, DELIPARASCHOS K M, MOUSTRIS G P. Fuzzy logic path tracking control for autonomous non-holonomic mobile robots: Design of System on a Chip[J]. *Robotics & Autonomous Systems*, 2010, 58(8): 1017-1027.

[65] RESENDE C Z, CARELLI R, SARCINELLI-FILHO M. A nonlinear trajectory tracking controller for mobile robots with velocity limitation via fuzzy gains[J]. *Control Engineering Practice*, 2013, 21(10): 1302-1309.

[66] ZOHAR I, AILON A, RABINOVICI R. Mobile robot characterized by dynamic and kinematic equations and actuator dynamics: Trajectory tracking and related application[J]. *Robotics & Autonomous Systems*, 2011, 59(6): 343-353.

[67] BLAZIC S. A novel trajectory-tracking control law for wheeled mobile robots[J]. *Robotics & Autonomous Systems*, 2011, 59(11): 1001-1007.

[68] HASHEMI E, JADIDI M G, JADIDI N G. Model-based PI-fuzzy control of four-wheeled omni-directional mobile robots[J]. *Robotics & Autonomous Systems*, 2011, 59(11): 930-942.

[69] SCAGLIA G, ROSALES A, QUINTERO L, et al. A linear-interpolation-based controller design for trajectory tracking of mobile robots[J]. *Control Engineering Practice*, 2010, 18(3): 318-329.

[70] MAUDER M. Robust tracking control of nonholonomic dynamic systems with application to the bisteerable mobile robot[J]. *Automatica*, 2008, 44 (10): 2588-2592.

[71] JIANGDAGGER Z P, NIJMEIJER H. Tracking control of mobile robots: a case study in backstepping[J]. *Automatica*, 1997, 33(7): 1393-1399.

[72] YOO S J. Adaptive neural tracking and obstacle avoidance of uncertain mobile robots with unknown skidding and slipping[J]. *Information Sciences An International Journal*, 2013, 238(7): 176-189.

[73] MOHARERI O, DHAOUADI R, RAD A B. Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks[J]. *Neurocomputing*, 2012, 88(7): 54-66.

[74] HENDZEL Z, SZUSTER M. Discrete neural dynamic programming in wheeled mobile robot control[J]. Communications in Nonlinear Science & Numerical Simulation, 2011, 16(5): 2355-2362.

[75] LIM H, KANG Y, KIM C, et al. Experimental verification of nonlinear model predictive tracking control for six-wheeled unmanned ground vehicles[J]. International Journal of Precision Engineering & Manufacturing, 2014, 15(5): 831-840.

[76] DINH V T, NGUYEN H, SHIN S M, et al. Tracking control of omnidirectional mobile platform with disturbance using differential sliding mode controller[J]. International Journal of Precision Engineering & Manufacturing, 2012, 13(1): 39-48.

[77] GONZALEZ R, FIACCHINI M, ALAMO T, et al. Adaptive control for a mobile robot under slip conditions using an LMI-based approach[C]//Control Conference. IEEE, 2009: 144-155.

[78] YOO S J. Adaptive neural tracking and obstacle avoidance of uncertain mobile robots with unknown skidding and slipping[J]. Information Sciences An International Journal, 2013, 238(7): 176-189.

[79] 袁国伟. 浅谈工业机器人在工业生产中的应用[J]. 科协论坛, 2013, (4): 58-59.

[80] 张栩. 面向室外移动机器人的路标定位系统[D]. 上海: 上海交通大学, 2010.

[81] 赵永翔. 基于无线局域网的室内定位系统研究[D]. 武汉: 武汉大学, 2010.

[82] 贾耀辉. 基于单目视觉的移动机器人目标定位方法研究[D]. 重庆: 重庆大学, 2010.

[83] 黄蓉, 刘敏. 基于 A* 算法求解最短路径的实现原理[J]. 企业家天地月刊, 2009, (7): 122-123.

[84] 杨海. 蚁群算法及其在智能交通中的应用[D]. 济南: 山东师范大学, 2008.

[85] 姜文波. 蚁群算法局部最优解决机制的探讨[J]. 智能计算机与应用, 2014, 4(3): 53-54.

[86] 徐德明. 改进的遗传混合蚁群算法在 TSP 问题中的应用[J]. 计算机时代, 2012, (11): 31-32.

[87] 施龙. 基于蚁群算法的分布式多智能体任务分配研究[D]. 杭州: 浙江工商大学, 2011.

[88] 毋玉芝, 周超. 蚁群算法及其在求解优化问题中的应用[J]. 科技信息: 科学教研, 2007, (31): 9.

[89] 袁梅. 基于蚁群算法的 WTA 问题求解[D]. 哈尔滨: 哈尔滨工业大学, 2007.

[90] 傅鹏. 多目标广义蚁群算法的收敛性、收敛速度和算法复杂度研究及其应用 [D]. 南京:南京邮电大学,2014.

[91] 王会颖. 蚁群算法及群体智能的应用研究 [D]. 合肥:安徽大学,2007.

[92] DORIGO M, CARO G D, STÜTZLE T. Ant algorithms [J]. Future Generation Computer Systems, 2000, 16(8): v-vii.

[93] 左敏,许华荣. 基于改进蚁群算法的智能小车路径规划 [J]. 心智与计算, 2011, 5 (2): 60-68.

[94] 何少佳,刘子扬. 基于惯性蚁群算法的机器人路径规划 [J]. 计算机工程与应用, 2012, 48 (15): 245-248.

[95] 付宁,张东霞,国海涛. 蚁群算法在机器人路径规划中研究及发展趋势 [J]. 潍坊工程职业学院学报, 2011, 24 (5): 81-82.

[96] 何娟,涂中英,牛玉刚. 一种遗传蚁群算法的机器人路径规划方法 [J]. 计算机仿真, 2010, 27 (3): 170-174.

[97] 李锋源,许艳萍,王武. 多策略蚁群算法求解机器人路径规划 [J]. 福州大学学报:自然科学版, 2011, 39 (3): 385-391.

[98] 薛小虎,南振岐,赵文杰. 基于改进的文化蚁群算法求解最优路径问题研究 [J]. 佳木斯大学学报:自然科学版, 2011, 29 (1): 54-57.

[99] VACCARO R J. Digital control [M]. New York: McGraw-Hill, 1995.

[100] KAZANTZIS, CHONG, Park, et al. Control-relevant discretization of nonlinear systems with time-delay using taylor-lie series [C]//Proceedings of the American Control Conference, 2003, 13: 149-154.

[101] Ji H P, CHONG K T, KAZANTZIS N, et al. Time-discretization of nonlinear systems with delayed multi-input using Taylor series [J]. Journal of Mechanical Science & Technology, 2004, 18(7): 1107-1120.

[102] YANG T T, LIU Z Y, CHEN H, et al. The research on robust tracking control of constrained wheeled mobile robots [C]//International Conference on Machine Learning and Cybernetics, 2005, 3: 1356-1361.

[103] TSAY D L, CHUNG H Y, LEE C J. The adaptive control of nonlinear systems using the sugeno-type of fuzzy logic [J]. Fuzzy Systems IEEE Transactions on, 1999, 7(2): 225-229.

[104] KAZANTZIS N, KRAVARIS C. Time-discretization of nonlinear control systems via Taylor methods [J]. Computers & Chemical Engineering, 1999, 23(6): 763-784.

[105] ZHANG Y L, PARK J B, CHONG K T. Controller design for nonlinear systems with time delay using model algorithm control (MAC) [J]. Simu-

lation Modelling Practice & Theory, 2009, 17(10): 1723-1733.

[106] CUI Y, LIU Y J, LI D J. Robust adaptive NN control for a class of uncertain discrete-time nonlinear MIMO systems[J]. Neural Computing & Applications, 2013, 22(3-4): 747-754.

[107] CHEN W S, JIAO L C, LI J, et al. Adaptive NN backstepping output-feedback control for stochastic nonlinear strict-feedback systems with time-varying delays[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 2009, 40(3): 939-950.

[108] LIU Y J, TONG S C, WANG D, et al. Adaptive neural output feedback controller design with reduced-order observer for a class of uncertain nonlinear SISO systems[J]. IEEE Transactions on Neural Networks, 2011, 22(8): 1328-1334.

[109] QIU Z C, ZHANG X T, YE C D. Vibration suppression of a flexible piezoelectric beam using BP neural network controller[J]. 固体力学学报(英文版), 2012, 25(4): 417-428.

[110] HAN L, ZHANG Z Y. The application of immune genetic algorithm in main steam temperature of PID control of BP network[J]. Physics Procedia, 2012, 24(1): 80-86.

[111] NØRGAARD M, RAVN O E, POULSEN N K, et al. Neural networks for modelling and control of dynamic systems [M]. London: Springer-Verlag, 2000.

[112] CHEMACHEMA M. Output feedback direct adaptive neural network control for uncertain SISO nonlinear systems using a fuzzy estimator of the control error[J]. Neural Networks the Official Journal of the International Neural Network Society, 2012, 36(3): 25-34.

[113] YEH M F. Single-input CMAC control system[J]. Neurocomputing, 2007, 70(16-18): 2638-2644.

YIDONG JIQIREN DAOHANG YU KONGZHI
SUANFA SHEJI



策划编辑 王 剑
责任编辑 张少奇
封面设计 原色设计

ISBN 978-7-5680-2448-1



9 787568 024488 >

定价：38.00元